General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

16783

LOCKHEED

JSC-[redacted]

ENGINEERING AND MANAGEMENT SERVICES COMPANY, INC.

OCT 0 6 1980

Ref: 644-1788
Contract NAS 9-15800
Job Order: 74-402

TECHNICAL MEMORANDUM

PURE PIXEL CLASSIFICATION SOFTWARE

By

O. A. Wehmanen

Approved By: Mike Pore

M. D. Pore, Supervisor
Accuracy Assessment
Section

July 1980

LEMSCO-15309

1830 NASA ROAD 1, Houston, Texas 77058 • Tel (713) 333-5411

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
JSC-16783		

4. Title and Subtitle	5. Report Date
Pure Pixel Classification Software	July 1980
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
O. A. Wehmanen	LEMSCO-15309
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Lockheed Engineering and Management Services Co., Inc. 1830 NASA Road 1 Houston, Texas 77058	NAS 9-15800
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 Technical Monitor: R. O. Hill/SF4	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

In this memorandum programs are described which permit classification runs with the LARSYS software to be made on images which have the ground truth field boundaries removed.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Pixel Clustering Classification Mixed Pixel LandSat Ag Survey	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified		

*For sale by the National Technical Information Service, Springfield, Virginia 22161

CONTENTS

1. INTRODUCTION

It has been hypothesized that boundary pixels, the so-called mixels, are a major source of classification error in the various clustering and classification algorithms applied to LANDSAT data. This classification error is due to (1) the distortion of the statistics for the classes identified by the algorithm caused by the inclusion of different targets and (2) because the label assignment for nonhomogeneous areas is not well defined. It is expected that if the boundary pixels were removed, the accuracy of clustering and classification would be greatly improved.

This document describes programs which generate an image file that has all ground truth boundary pixel spectral values set to one value. This image, when processed by LARSYS routines, gives classification and clustering maps with all boundary pixels assigned to one class.

Using these programs the performance of clustering and classification procedures for pure pixels can be tested.

2. DESCRIPTION OF THE SOFTWARE

The ground truth data are available at JSC, thus the ground truth processing takes place on the image processor in the Data Techniques Laboratory. The spectral data are available both at JSC and LARS. Since the clustering and classification system is included in LARSYS at Purdue, the ground truth purity data have been merged with the spectral data at LARS. The flow of data is shown in figure 1.

2.1 Image Processor Programs

2.1.1 GROUND TRUTH INPUT

The ground truth input comes from disk files installed by accuracy assessment software on a disk mounted on the second disk drive (DB2) of the image processor. This input is a digital map generated from ground truth data with six pixels for each LANDSAT pixel. These data are documented in "Format

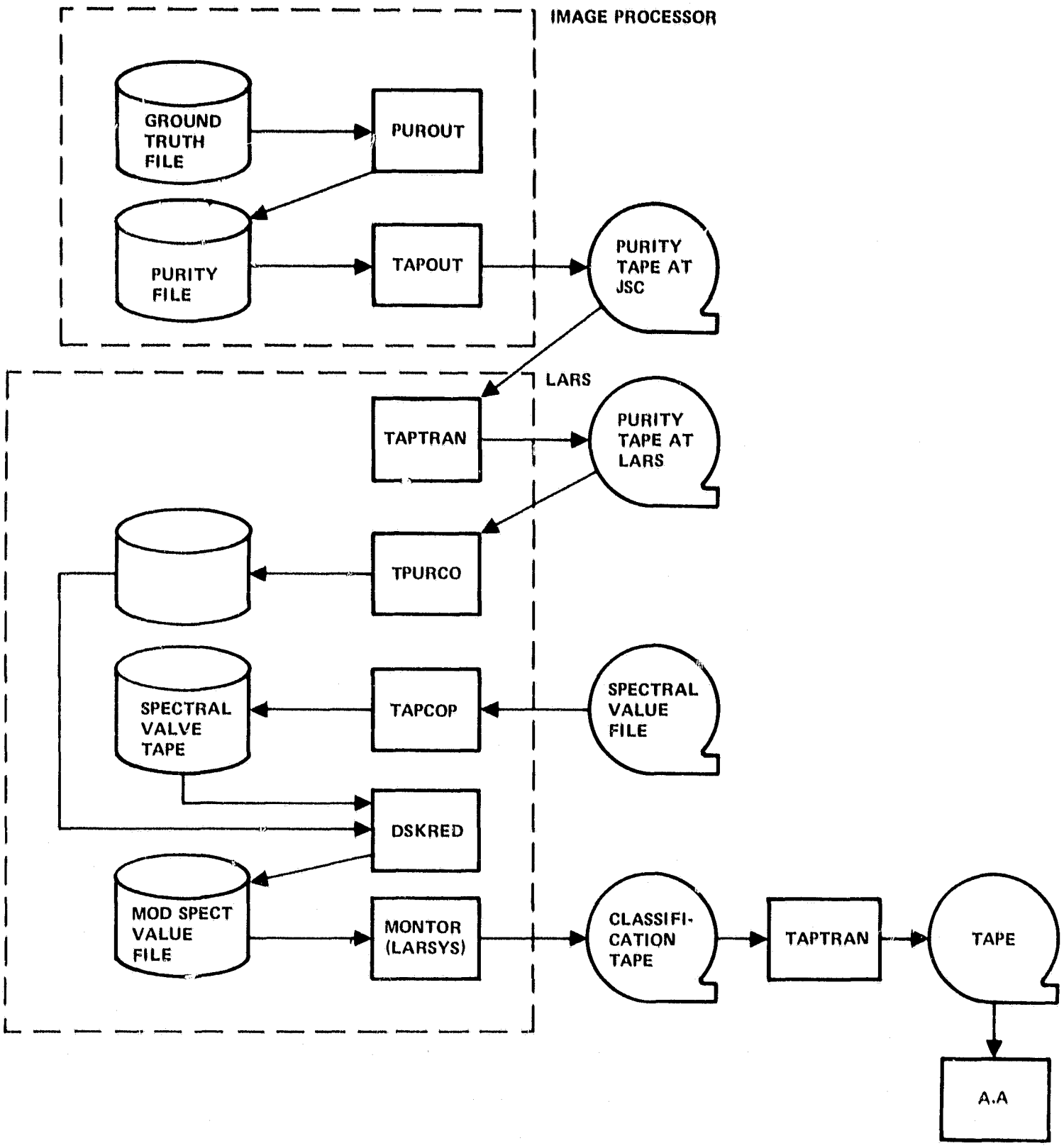


Figure 1.- Data flow.

Specifications for LACIE (Phase III) and Accuracy Accessment Computer Data Products," LEMSCO-12507.

2.1.2 PIXEL PURITY IMAGE PROGRAM (PUROUT)

2.1.2.1 Linkage

PUROUT calls subroutine PURE at the entry points PURE, PURE1, PURE2 and ROLL. PUROUT calls subroutine STRIP at the entry points STIN and STRIP.

2.1.2.2 Interface

PUROUT communicates with PURE and STRIP through the common block /PURE/.

2.1.2.3 Input

PUROUT reads FILNAM.DAT to get the name of the image data file on logical unit 2. PUROUT reads the image file for example DB2:[111,3]013579999.GTO, on Logical unit 1.

2.1.2.4 Output

PUROUT writes the image file of type PT1 on logical unit 3. For example: [111,3]013579999.PT1).

2.1.2.5 Storage

Total space allocated 3184.

2.1.2.6 Description

PUROUT first reads the input file name and checks it interactively with the operator. Then the header is copied, unchanged, to the output file and processing begins.

PUROUT holds three lines of data in the array BUF (392,9). Since each pixel corresponds to 6 subpixels each line occupies a 392 x 3 block of space. The data are read into sublines 7, 8, and 9. Subroutine PURE determines whether the pixels on the input line are pure. Pure means that all subpixels are in

the same class. Subroutine ROLL moves all the data up one line. Subroutine PURE1 checks pure pixels in 4, 5, and 6 and marks those surrounded by subpixels of the same class as "more-pure". PURE2 checks pure pixels in 4, 5, and 6 and marks those surrounded by pure pixels of the same class as "most-pure". STRIP removes strip-fallow classes. As data are rolled to the top of the array it is written to the output file.

2.1.3 SUBROUTINE PURE

2.1.3.1 Linkage

PURE has four entry points, PURE, PURE1, PURE2, and ROLL.

2.1.3.2 Interface

All information is transferred through the common block/PURE/. This block contains the input Byte data, array, BUF (392,9), and the output integer *2 array LAB (196,3).

2.1.3.3 Input

No input

2.1.3.4 Output

No output

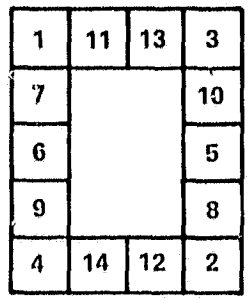
2.1.3.5 Storage

Total space allocated 2729

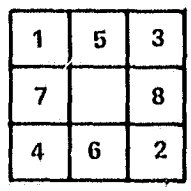
2.1.3.6 Description

1. Entry PURE — A pixel is pure only if all subpixels are of the same subclass. Subroutine PURE marks pure pixels with "1" and impure pixels with "0". The majority label is also saved. PURE works on the bottom line, sublines 7, 8, and 9 of the input array.

2. Entry PURE1 — PURE1 checks pure pixels in line 2 and sublines 4, 5, and 6. The purity label is changed to "2" if all neighboring subpixels have the same label as the pixel. The figure below shows the order of checking.



3. Entry PURE2 — PURE2 checks more pure (Label = "2") pixels in line 2 and sublines 4, 5, and 6, and changes the label to "3" if all adjacent pixels have the same label. Pixels are checked in the order shown in the figure below.



4. Entry ROLL — ROLL moves the data up one line and three sublines, in preparation for new input. Line 1 and sublines 1, 2, and 3 are destroyed in the process. Sublines 7, 8, and 9 are not cleared.

2.1.4 SUBROUTINE STRIP

2.1.4.1 Linkage

STRIP has two entry points, STRIP and STIN.

2.1.4.2 Interface

STRIP transfers data through the common block /PURE/ and through the calling arguments.

2.1.4.3 Input

Entry STIN reads the array ZAP (256) on logical unit 8 from the file specified in the array CRDFIL (32).

2.1.4.4 Output

STIN may type an error message.

2.1.4.5 Storage

Total space allocated 2778

2.1.4.6 Description

1. STRIP (KK)

   STRIP changes the purity class of marked classes to "0". The marked classes are typically problem fields, strip fields, and non-inventoried fields. These fields can be marked in two different ways. Three different sets of class identifiers may be coded into the array TEST (8, 3), or the array ZAP (256) may be read in. If KK = 10, the ZAP alternative is used. If KK = 1, 2, or 3 the array TEST (*, KK) is used.

2. STIN (CRDFIL)

   STIN reads the array ZAP from the file specified by CRDFIL. If ZAP (N) = 0, class N is accepted, if ZAP (N) = 1, class N is marked and purity will be set to "0".

2.1.5 TAPE GENERATION (TAPEOUT)

The program TAPEOUT outputs a universal format tape from a disk file (reference Action Document, 63-3107-4402-16).

2.1.6 TASKBUILDER COMMAND FILE

The file PUROUT.CMD contains the taskbuilder commands needed to construct the PUROUT.TSK file.

2.2 LARS PROGRAMS

2.2.1 TAPE TRANSFER (TAPTRAN)

TAPTRAN is a program written and maintained by Purdue. It is documented in "LARS DATA - 100 Operator's Manual."

2.2.2 PURITY IMAGE TAPE TO DISK (TPURCO)

2.2.2.1 Linkage

None

2.2.2.2 Interface

None

2.2.2.3 Input

TPURCO reads a universal format, 1 channel tape from unit 11. The line size is 90 INTEGER*4 words or 360 Bytes.

2.2.2.4 Output

The header is copied to unit 13. The data are copied to unit 12. A small report is written on unit 6.

2.2.2.5 Storage

Program Size = 4050.

2.2.2.6 Description

TPURCO reads the tape and copies it to disk files.

2.2.3 SPECTRAL VALUE TAPE TO DISK (TAPCOP)

2.2.3.1 Linkage

TAPCOP calls GETACQ, RTEERR, and TOPRD. These are all Purdue maintained routines. Documentation can be found in LARS Program abstract 11 for module TAPOP and LARS Program Abstract 2020 for module GTINFO.

2.2.3.2 Interface

Interface is through the calling arguments and the tape mounted by GETACQ.

2.2.3.3 Input

TAPCOP interactively gets the segment name and date and TOPRD reads the tape mounted by GETACQ.

2.2.3.4 Output

The header of the universal format input tape is written on unit 13. The 4-channel spectral data are written on unit 12.

2.2.3.5 Storage

Program size = 4822.

2.2.3.6 Description

TAPCOP interactively gets segment and data. These are passed to GETACQ which mounts the correct LARS library tape and positions it at the correct file. RTEERR decodes the error flag returned by GETACQ. If there is no error, TOPRD reads the tape which is then written to a disk file for further processing.

2.2.4 FILE MERGE (DSKRED)

2.2.4.1 Linkage

DSKRED calls TRNSLT.

2.2.4.2 Interface

The interface is through the calling arguments.

2.2.4.3 Input

DSKRED interactively gets the desired purity class from the terminal. A universal header is read from unit 17. Four-channel spectral data are read from unit 18 and 1-channel purity data are read from unit 19.

2.2.4.4 Output

A universal format image tape file is written on unit 20.

2.2.4.5 Storage

Program size = 9918.

2.2.4.6 Description

DSKRED reads the spectral values and purity values. For those pixels with less purity than is desired, the spectral values are changed to ch 1 = 0, ch 2 = 0, ch 3 = 0, and ch 4 = 255.

Then these data are written out in universal format.

2.2.5 BYTE MANIPULATION (TRNSLT)

2.2.5.1 Linkage

Subroutine TRNSLT does not call any other program.

2.2.5.2 Interface

All data are passed through the calling arguments.
TRNSLT (DUF, PUF, DH1, DH2, DH3, DH4, PH, OPTION)
DUF (225) spectral value input line
PUF (90) purity value input line
DH1 (196) - DH4 (196) output values
PH (196) purity output values
OPTION. If OPTION = 1 DH
is KAUTH transformed
If OPTION = 2 DH
is LANDSAT 3 corrected
and KAUTH transformed
If OPTION = 0 DH is raw channel values.

2.2.5.3 Input

None

2.2.5.4 Output

None

2.2.5.5 Storage

Program size = 8202.

2.2.5.6 Description

Subroutine TRNSLT converts one line of spectral data in bytes to four-integer arrays, and also one line of purity data in bytes to an integer array. The spectral output may be raw channel values, KAUTH transformed values or Lockheed/EMSCO's LANDSAT 3 corrected KAUTH transformed values.

The data are placed in LOGICAL*1 arrays by equivalence statements and then assigned to integer arrays.

2.2.6 EXECUTIVE ROUTINES

For the programs TPURCO TAPCOP and DSKRED there are EXEC files with the same names which give the required FILEDEF commands and start execution. In addition RTE EXEC may be executed to give the needed GETDISK commands. Subroutines GETACQ, RTEERR, and TOPRD reside on JSC19A.

2.3 MODIFICATIONS TO LARSYS ROUTINES

To run LARSYS on the output file of DSKRED the supervisor program, MONTOR, is used. Also, three of the LARSYS subroutines required slight modification. These modified programs reside on JSC808.

2.3.1 WRTHED

The information saved by LARSYS for the output tape header was deemed inadequate. Therefore code was added to read the input file header and write it to the output file after two small changes were made.

WRTHED calls the new subroutine IDTE for current date.

2.3.2 PSPPAT

Because the channel 4 value for impure pixels is set to 255, the accumulator register for sum of channel 4 squared has excessive error. It was necessary to change this variable to REAL*8, double precision to avoid excessive error.

2.3.3 COVPAT

Because the values for all impure pixels are the same, the covariance matrix for impure pixels is singular. LARSYS rejects singular covariance matrices. Code was modified in subroutine COVPAT to insert a nonsingular covariance matrix whenever mean channel 4 exceeds 250 counts.

2.3.4 CURRENT DATE (IDTE)

2.3.4.1 Linkage

IDTE calls GTDATE and IMONTH.

2.3.4.2 Interface

All interface is through the calling arguments.
NA is month number
NB is day of month
NC is year.
For the system routine GTDATE the array data are in A format. For example, printing DATE as 3A4 gives June 14, 1980.

2.3.4.3 Input

None.

2.3.4.4 Output

None.

2.3.4.5 Storage

Program Size = 524.

2.3.4.6 Description

Subroutine IDII obtains the current date in Alphanumeric format. Conversion of the numbers is done by writing and rereading. IMONTH is called to convert the month.

2.3.5 MONTH CONVERSION (IMONTH)

2.3.5.1 Linkage

None.

2.3.5.2 Interface

Date is a 4-character month name. I is the integer month number.

2.3.5.3 Input

None.

2.3.5.4 Output

Possible error statement on unit 6.

2.3.5.5 Storage

Program size = 496.

2.3.5.6 Description

IMONTH compares the month name to test values until a match is found. If no match is found an error statement is generated.

3. LISTINGS

```
FORTRAN IV+PLUS V02-51          11:48:30    30-JUN-80          PAGE 1
PURBUT.FTN      /TR:PL2CKS/WR

      C  PURBUT.FTN
      C        THIS PROGRAM GENERATES PIXEL PURITY MAPS
      C        '1' MEANS PURE
      C        '4' MEANS PURE WITH A 1 SUBPIXEL PURE HALO
      C        '2' MEANS PURE WITH A 1 PIXEL PURE HALO
      C        '3' MEANS IMPURE
      C        '0' MAY ALSO MEAN IMPURE
      C
      C
0001            IMPLICIT INTEGER*2  (A-Z)
0002            INTEGER*2 LAB(196,3)
0003            BYTE      BUF(392,9),BYL(196,2,3),DUM(72)
0004            BYTE      RDIN1(392),RDIN2(392),RDIN3(392)
0005            BYTE      LBUT(196),FILNAM(32),HEADER(3060)
0006            BYTE      FILL(92),CRDFIL(32)
      C
0007            COMMON /PURE/ BUF,LAB
0008            EQUIVALENCE (BYL(1,1,1),LAB(1,1))
0009            EQUIVALENCE (BYL(1,2,1),LBUT(1))
0010            EQUIVALENCE (BUF(1,1),HEADER(1))
0011            EQUIVALENCE (RDIN1(1),BUF(1,7))
0012            EQUIVALENCE (RDIN2(1),BUF(1,8))
0013            EQUIVALENCE (RDIN3(1),BUF(1,9))
0014            DATA  FILL/92*0/
0015            DATA  CRDFIL/'C','R','D','F','I','L',',',',D','A',
     *                  'T',22*0/
      C
      C
0016            CALL STRIN(CRDFIL)
0017            OPEN(UNIT=2,NAME='FILNAM.DAT',TYPE='OLD',
     *          READONLY,ACCESS='SEQUENTIAL',ERR=24)
0018      155     CONTINUE
0019            READ(2,22,END=166) FILNAM
0020       22   FORMAT(32A1,2X,I4)
0021       24   CONTINUE
0022            LINES=117
0023            ZI=1
0024            ZIP=1
0025            DO 21 I=1,32
0026            IF(FILNAM(I).EQ.'.')ZDT=0
0027            IF(FILNAM(I).EQ.',')ZIP=I
0028       21   IF(ZDT.EQ.0)FILNAM(I)=0
0029            TYPE 23, FILNAM,LINES
0030       23   FORMAT('    FILNAM ',32A1,'   LINES = ',I6,
     *          /,'    S TO STOP      CR TO CONTINUE')
0031            ACCEPT 25,I25
0032       25   FORMAT(A1)
0033            IF(I25 .EQ. 'S') STOP    ! I25 !
      C
      C
0034            OPEN(UNIT=1,NAME=FILNAM,TYPE='OLD',ACCESS='SEQUENTIAL'
     *          ,READONLY,FORM='UNFORMATTED',ERR=666)
0035            FILNAM(ZIP+1)='P'
0036            FILNAM(ZIP+2)='T'
0037            FILNAM(ZIP+3)='1'
0038            OPEN(UNIT=3,NAME=FILNAM,TYPE='NEW'
```

```
     *          ,ACCESS='SEQUENTIAL',FORM='UNFORMATTED',ERR=666)
      C
      C
0039              GO TO 888
0040    666       WRITE(6,66) FILNAM
0041    66        FORMAT('    OPEN ERROR UNIT 1; FILNAM = ',24A1)
0042              STOP ' 666   OPEN ERROR FILE 1'
0043    888       CONTINUE
0044              READ(1) HEADER
0045              WRITE(3) HEADER
0046              READ(1) DUM,RDIN1
0047              READ(1) DUM,RDIN2
0048              READ(1) DUM,RDIN3
0049              CALL PURE
0050              CALL ROLL
0051              READ(1) DUM,RDIN1
0052              READ(1) DUM,RDIN2
0053              READ(1) DUM,RDIN3
0054              CALL PURE
0055              CALL ROLL
0056              READ(1) DUM,RDIN1
0057              READ(1) DUM,RDIN2
0058              READ(1) DUM,RDIN3
0059              CALL PURE
0060              CALL PURE1
0061              CALL PURE2
      D           LINES = 12
0062              CALL STRIP(10)
0063              WRITE(3) DUM,LOUT,FILL
0064              DO 100 I=4,LINES
0065              CALL ROLL
0066              READ(1) DUM,RDIN1
0067              READ(1) DUM,RDIN2
0068              READ(1) DUM,RDIN3
0069              CALL PURE
0070              CALL PURE1
0071              CALL PURE2
      D           WRITE(6,96) LOUT
0072    96        FORMAT('  LOUT  ',49A2)
0073              CALL STRIP(10)
0074              WRITE(3) DUM,LOUT,FILL
      D           WRITE(6,97)BUF
0075    97        FORMAT('  BUF   ',98A1)
0076    100       CONTINUE
0077              CALL ROLL
0078              CALL STRIP(10)
0079              WRITE(3) DUM,LOUT,FILL
0080              CALL ROLL
0081              CALL STRIP(10)
0082              WRITE(3) DUM,LOUT,FILL
0083              CALL ROLL
0084              CALL STRIP(10)
0085              WRITE(3) DUM,LOUT,FILL
0086              CLOSE(UNIT=1)
0087              CLOSE(UNIT=3)
0088              GO TO 155
```

```
FØRTRAN IV+PLUS V02+51          11:48:30    30+JUN+80          PAGE 3
PURØUT,FTN      /TABLEØCKS/WR

0089    166     STØP    'END ØF DATA UNIT 2'
0090            ENØ
```

```
FORTRAN IV-PLUS V02-51          11:48:30    30-JUN-80          PAGE 4
PURSUT.FTN      /TABLOCKS/WR

 PROGRAM SECTIONS

 NUMBER   NAME      SIZE              ATTRIBUTES

   1      SCODE1  002060   536        RW,I,GBL,LCL
   2      SPDATA  001104    34        RW,D,GBL,LCL
   3      SIDATA  001436   143        RW,D,GBL,LCL
   4      SVARS   000356   119        RW,D,GBL,LCL
   6      PURE    011140  2352        RW,I,RVR,GBL

 VARIABLES

 NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS

 I     I*2   4-000352   I25   I*2   4-000354   LINES I*2   4-000344   ZIP   I*2   4-000350   ZBT   I*2   4-000346

 ARRAYS

 NAME    TYPE  ADDRESS      SIZE        DIMENSIONS

 BUF     L*1   6-000000  006710  1764   (392,9)
 BYL     L*1   6-002710  002230   568   (196,2,3)
 CRDFIL  L*1   4-001304  000040    16   (32)
 DUM     L*1   4-001000  000110    36   (72)
 FILL    L*1   4-001150  000134    46   (92)
 FILNAM  L*1   4-001110  000040    16   (32)
 HEADER  L*1   6-001000  005764  1530   (3060)
 LAB     I*2   6-002710  002230   568   (196,3)
 LOUT    L*1   6-002214  000304    98   (196)
 RDIN1   L*1   6-005460  000610   196   (392)
 RDIN2   L*1   6-005270  000610   196   (392)
 RDIN3   L*1   6-005100  000610   196   (392)

 LABELS

 LABEL  ADDRESS      LABEL  ADDRESS      LABEL  ADDRESS      LABEL  ADDRESS      LABEL  ADDRESS

 21     **           22'    3-000000     23'    3-000010     24     1-000112     25'    3-000114
 66'    3-000120     96'    **           97'    **           100    **           155    1-000046
 166    1-002042     666    1-000406     888    1-000454

 FUNCTIONS AND SUBROUTINES REFERENCED

 CLOSS  OPENS  PURE   PURE1  PURE2  RDLL  STIN  STRIP

 TOTAL SPACE ALLOCATED = 014340  3184

 NO FPP INSTRUCTIONS GENERATED

 PURSUT,LP:=PURSUT
```

```
FORTRAN IV-PLUS V02-51          08-12-17    01-JUL-80          PAGE 1
PURE.FTN        /TR:BL2CKS/WR

                C  PURE.FTN
                C
                C          PURE IS SUBR2UTINES FOR A:A1 PURE PIXEL TEST
                C
                C
0001                    SUBR?UTINE PURE
0002                    IMPLICIT INTEGER*2 (A-Z)
0003                    INTEGER*2 LAB(196,3)
0004                    BYTE      BUF(392,9),BYL(196,2,3)
0005                    INTEGER*2 DEL1(6),DEL2(6),DEL10(14),DEL11(14),DEL20(8),DEL21(8)
                C
                C
0006                    COMMON  /PURE/ BUF,LAB
0007                    EQUIVALENCE (BYL(1,1,1),LAB(1,1))
0008                    DATA DEL1/0,0,0,1,1,1/, DEL2/7,8,9,7,8,9/
0009                    DATA DEL10/-1,2,2,-1,2,-1,-1,2,-1,2,0,1,1,0/
0010                    DATA DEL11/ 3,7,3, 7,5, 5, 4,6, 6,4,3,7,3,7/
0011                    DATA DEL20/-1,1,1,-1,0,0,-1,1/
0012                    DATA DEL21/ 1,3,1, 3,1,3, 2,2/
                C
                C
                C          PURE SETS A MAJORITY RULE LABEL FOR LAST ROW
                C
                C
                D       WRITE(6,145) LAB(1,1),LAB(1,2),LAB(1,3),BUF(1,1),
                D      *             BUF(1,2),BUF(1,3),BUF(1,4),BUF(1,5),
                D      *             BUF(1,6),BUF(1,7),BUF(1,8),BUF(1,9)
0013             145    FORMAT('  PURE  ',12A4,//)
0014                    DO 100  I=1,196
0015                    J=2*I-1
0016                    K=0
0017                    NN='1'
                C       'O' MEANS IMPURE  '1' IS PURE
0018                        J1=J+DEL1(1)
0019                        J2=DEL2(1)
0020                    DO 102 KK=2,6
0021                        K1=J+DEL1(KK)
0022                        K2=DEL2(KK)
0023             102        IF(BUF(J1,J2) .NE. BUF(K1,K2)) GO TO 103
0024                         GO TO 101
0025             103         NN='0'
0026             101         CONTINUE
0027                    BYL(I,1,3)=BUF(J1,J2)
0028                    BYL(I,2,3)=NN
0029             100    CONTINUE
                D       WRITE(6,145) LAB(1,1),LAB(1,2),LAB(1,3),BUF(1,1),
                D      *             BUF(1,2),BUF(1,3),BUF(1,4),BUF(1,5),
                D      *             BUF(1,6),BUF(1,7),BUF(1,8),BUF(1,9)
0030                    RETURN
                C
                C
                C          PURE SETS LABEL =0 FOR IMPURE =1 FOR PURE
                C          PURE1 WILL ADD 1 FOR ADDITIONAL PURE HALO
                C               PURE PLUS HALO GIVES 92
                C
                C
0031                    ENTRY PURE1
```

```
      C
      D       WRITE(6,245) LAB(I,1),LAB(I,2),LAB(I,3),BUF(I,1),
      D     *              BUF(I,2),BUF(I,3),BUF(I,4),BUF(I,5),
      D     *              BUF(I,6),BUF(I,7),BUF(I,8),BUF(I,9)
0032   245    FORMAT(' PURE1      ',12A5,//)
      C
0033          DO 200 I=2,195
0034          IF(BYL(I,2,2) .NE. '1') GO TO 200
0035          Y=2.*I-1
0036          DO 201 KJ=1,14
0037          Y1=J+DEL10(KJ)
0038          Y2=DEL11(KJ)
0039          IF(BYL(I,1,2) .NE. BUF(Y1,Y2)) GO TO 200
0040   201    CONTINUE
0041          BYL(I,2,2)='2'
0042   200    CONTINUE
      D       WRITE(6,245) LAB(I,1),LAB(I,2),LAB(I,3),BUF(I,1),
      D     *              BUF(I,2),BUF(I,3),BUF(I,4),BUF(I,5),
      D     *              BUF(I,6),BUF(I,7),BUF(I,8),BUF(I,9)
0043          RETURN
      C
      C
      C       PURE2 CHECKS ADJACENT PIXELS FOR A HALO OF PURE PIXELS
      C
      C
0044          ENTRY PURE2
      C
      D       WRITE(6,345) LAB(I,1),LAB(I,2),LAB(I,3),BUF(I,1),
      D     *              BUF(I,2),BUF(I,3),BUF(I,4),BUF(I,5),
      D     *              BUF(I,6),BUF(I,7),BUF(I,8),BUF(I,9)
0045   345    FORMAT(' PURE2      ',12A5,//)
0046          DO 300 I=2,195
0047          IF(BYL(I,2,2) .NE. '2') GO TO 300
0048          DO 301 KJ=1,8
0049          I1=I+DEL20(KJ)
0050          I2=DEL21(KJ)
0051        IF(BYL(I1,2,I2).LT.'1' .OR. BYL(I1,2,I2).GT.'3')
     *        GO TO 300
0052        IF(BYL(I,1,2).NE.BYL(I1,1,I2)) GO TO 300
0053   301    CONTINUE
0054          BYL(I,2,2)='3'
0055   300    CONTINUE
      D       WRITE(6,345) LAB(I,1),LAB(I,2),LAB(I,3),BUF(I,1),
      D     *              BUF(I,2),BUF(I,3),BUF(I,4),BUF(I,5),
      D     *              BUF(I,6),BUF(I,7),BUF(I,8),BUF(I,9)
0056          RETURN
      C
      C
      C       ROLL  MOVES ALL THE DATA UP 1 LINE, (3 SUBLINES)
      C
      C
0057          ENTRY ROLL
      C
      D       WRITE(6,445) LAB(I,1),LAB(I,2),LAB(I,3),BUF(I,1),
      D     *              BUF(I,2),BUF(I,3),BUF(I,4),BUF(I,5),
      D     *              BUF(I,6),BUF(I,7),BUF(I,8),BUF(I,9)
```

```
        C
0058                DØ 400 I=1,196
0059                LAB(I,1)=LAB(I,2)
0060     400        LAB(I,2)=LAB(I,3)
0061                DØ 401 J=1,392
0062                LK=3
0063                DØ 401 K=1,6
0064                LK=LK+1
0065                BUF(J,K)=BUF(J,LK)
0066     401        CONTINUE
        D           WRITE(6,445) LAB(1,1),LAB(1,2),LAB(1,3),BUF(1,1),
        D     *                  BUF(1,2),BUF(1,3),BUF(1,4),BUF(1,5),
        D     *                  BUF(1,6),BUF(1,7),BUF(1,8),BUF(1,9)
0067     445        FØRMAT(' RØLL ',3A4,2X,9A3,///)
0068                RETURN
        C
        C
        C
        C
0069                END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	SCODE1	001146	307	RW,I,CON,LCL
4	SVARS	000214	70	RW,D,CON,LCL
6	PURE	012140	2352	RW,I,REL,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
PURE		1-000000	PURE1		1-000314	PURE2		1-000526	RBLL		1-000746			

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
I	I*2	4-000160	I1	I*2	4-000204	I2	I*2	4-000206	J	I*2	4-000162	J1	I*2	4-000170
J2	I*2	4-000172	K	I*2	4-000212	KJ	I*2	4-000202	KK	I*2	4-000174	K1	I*2	4-000176
K2	I*2	4-000200	LK	I*2	4-000210	N	I*2	4-000164	NN	I*2	4-000166			

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
BUF	L*1	6-000000	006710	1736	(392,9)
BVL	L*1	6-000710	002230	588	(196,2,3)
DEL1	I*2	4-000000	000014	6	(6)
DEL10	I*2	4-000030	000034	14	(14)
DEL11	I*2	4-000064	000034	14	(14)
DEL2	I*2	4-000014	000014	6	(6)
DEL20	I*2	4-000120	000020	8	(8)
DEL21	I*2	4-000140	000020	8	(8)
LAB	I*2	6-002710	002230	588	(196,3)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
100	**	101	1-000234	102	**	103	1-000220	145'	**
200	1-000502	201	**	245'	**	300	1-000722	301	**
345'	**	400	**	401	**	445'	**		

TOTAL SPACE ALLOCATED = 012522 2729

NO FPP INSTRUCTIONS GENERATED

```
FORTRAN IV-PLUS V02-51          11:49:14    30-JUN-80          PAGE 1
STRIP.FTN       /T:18LOCKS/WR

      C  STRIP.FTN
      C       IDENTIFIES STRIP, NONINVENTORIED, PROBLEM FIELDS
      C       AND SETS THEIR PURITY TO 0
      C
      C
0001          SUBROUTINE STRIP(KK)
      C
      C
0002          IMPLICIT INTEGER*2 (A-Z)
0003          INTEGER*2 LAB(196,3),TT
0004          BYTE      BUF(392,9),BYL($98,2,3)
0005          BYTE      TEST(8,3),ZAP(256)
0006          BYTE      CRDFIL(32)
      C
      C
0007          COMMON  /PURE/ BUF,LAB
0008          EQUIVALENCE (BYL(1,1,1),LAB(1,1))
0009          DATA IID/0/
0010          DATA TEST/80,                 !  PROBLEM FIELD
     *              164,                    !  NON INVENTORIED
     *              165,185,190,210,215,235, !  STRIP FIELD
     *              16*0/                   !  FUTURE CODES
      C
0011          DATA ZAP /79*0,1,             !  PROBLEM FIELD
     *              83*0,1,                 !  NON INVENTORIED
     *              21*1,4*0,21*1,4*0,21*1, !  STRIP FIELDS
     *              21*0/
      C
0012          IF(KK .EQ. 10) GO TO 200
0013          DO 100  I=1,196
0014          IF(  BYL(I,2,1) .LT. '1'
     *    .OR. BYL(I,2,1) .GT. '3') GO TO 100
0015          TT=BYL(I,1,1)
0016          TT=TT*128
0017          IF(TT .EQ. TEST(1,KK)) BYL(I,2,1) = '0'
0018          IF(TT .EQ. TEST(2,KK)) BYL(I,2,1) = '0'
0019          IF(TT .GE. TEST(3,KK)
     *    .AND. TT .LE. TEST(4,KK)) BYL(I,2,1) = '0'
0020          IF(TT .GE. TEST(5,KK)
     *    .AND. TT .LE. TEST(6,KK)) BYL(I,2,1) = '0'
0021          IF(TT .GE. TEST(7,KK)
     *    .AND. TT .LE. TEST(8,KK)) BYL(I,2,1) = '0'
0022      100 CONTINUE
0023          RETURN
      C
      C
0024      200 CONTINUE
      D       IF(IID.NE.1) WRITE(6,222) ZAP
0025      222 FORMAT( 5(I    ',10[2,2X))
      D       IID=1
0026          DO 201 I=1,196
0027          TT = BYL(I,1,1)
0028          TT=TT*128
0029          IF(ZAP(TT) .EQ. 1) BYL(I,2,1) = '0'
0030      201 CONTINUE
0031          RETURN
```

```
      C
      C
0032          ENTRY STIN(CRDFIL)
      C                          READ ZAP ARRAY FROM CARD FILE
      C
0033          OPEN(UNIT=8,NAME=CRDFIL,ACCESS='SEQUENTIAL',
     *          TYPE='OLD',READONLY,ERR=444)
0034          READ(8,455) ZAP
0035     455  FORMAT(60I1)       ! 5 CARDS
0036          CLOSE(UNIT=8)
      D       WRITE(6,456) ZAP,CRDFIL
0037     456  FORMAT(4(' ZAP ',60I1,/),' ZAP ',14I1,
     *            ' STIN    FILE NAME = ',32A1)
0038          RETURN
      C
0039     444  TYPE 443,CRDFIL
0040     443  FORMAT(' ERROR OPENING UNIT 8 ',
     *         /' FILENAME = ',32A1,
     *         /' DEFAULT VALUES USED')
0041          RETURN
0042          END
```

```
FORTRAN IV-PLUS V02-51          11:49:14    30-JUN-80          PAGE 3
STRIP.FTN       /TABLEBOCKS/NR

PROGRAM SECTIONS

NUMBER   NAME       SIZE                ATTRIBUTES

  1      $CODE1   000672   221          RW,I,LCL,LCL
  3      $IDATA   000174    62          RW,D,CON,LCL
  4      $VARS    000436   143          RW,D,CON,LCL
  6      PURE     011140  2352          RW,D,SAV,GBL


ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 STIN         1=000502   STRIP        1=000000


VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 I      I*2   4=000434   IID    I*2   4=000436   KK     I*2   F=000002*  TT     I*2   4=000000


ARRAYS

 NAME   TYPE  ADDRESS       SIZE        DIMENSIONS

 BUF    L*1   6=000000  006710  1764   (392,9)
 BYL    L*1   6=000710  002230   588   (196,6,3)
 CRDFLL L*1   F=000002* 000040    16   (32)
 LAB    I*2   6=000710  002230   588   (196,3)
 TEST   L*1   4=000002  000030    12   (6,3)
 ZZZ    L*1   4=000032  000400   128   (256)


LABELS

 LABEL  ADDRESS         LABEL  ADDRESS         LABEL  ADDRESS         LABEL  ADDRESS         LABEL  ADDRESS

 100    1=000364        200    1=000410        201      **            222'     **            443'   3=000004
 444    1=000612        455'   3=000000        456'     **


FUNCTIONS AND SUBROUTINES REFERENCED

 CLOSS   OPENS


TOTAL SPACE ALLOCATED = 012664  2778

NO FPP INSTRUCTIONS GENERATED

STRIP,LP1=STRIP
```

FILE TPURCO PURDUE / LARS 3031

```
                C                                                                TPU00010
                C  TPURCO FFORTRAN          COPIES PURITY TAPES TO DISK          TPU00020
                C                                                                TPU00030
0001                  INTEGER   BUF(7??),LINE(90)                                TPU00040
0002                  READ (11,11,END=226) BUF                                   TPU00050
0003                  WRITE (13,11) BUF                                          TPU00060
0004             11   FORMAT(17(4?A4,/))                                         TPU00070
0005             226  CONTINUE                                                   TPU00080
                C                                                                TPU00090
0006                  DO 221 I=1,11?                                             TPU00100
0007                  READ(11,12,END=555) LINE                                   TPU00110
0008                  WRITE(12,12) LINE                                          TPU00120
0009             12   FORMAT(6(15A4))                                            TPU00130
0010                  IF(I .LT. 4) WRITE(6,76)                                   TPU00140
0011             76   FORMAT(//,'   LINE')                                       TPU00150
0012                  WRITE(6,77) LINE                                           TPU00160
0013             77   FORMAT(' ',1?I10)                                          TPU00170
0014             221  CONTINUE                                                   TPU00180
                C                                                                TPU00190
0015             555  CONTINUE                                                   TPU00200
0016                  STOP                                                       TPU00210
0017                  END                                                        TPU00220
```

FILE TPURCO PROBLEM / LABS 10.31

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IHCOM#	A8								

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
I	AC								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
BUF	B0	LINE	C84						

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
11	E0C	12	E15	76	E1E	77	E2B		

```
*OPTIONS IN EFFECT*   IOS,EBCDIC,SOURCE,NOLIST,DECK,NOLOAD,MAP
*OPTIONS IN EFFECT*   NAME = TPURCO  , LINECNT =      75
*STATISTICS*     SOURCE STATEMENTS =        17,PROGRAM SIZE =     3982
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

FILE TAPCOP PURDUE / LARS 3031

```
                C
                C  TAPCOP FFORTRAN                                                      TAP00010
                C                                                                       TAP00020
                C                                                                       TAP00030
                C                                                                       TAP00040
0001                  IMPLICIT INTEGER *4 (A-Z)                                         TAP00050
0002                  INTEGER   BUF(700),LINE(225)                                      TAP00060
                C                                                                       TAP00070
0003                  WRITE (16,20)                                                     TAP00080
0004             20   FORMAT  (' ENTER SEG AND DATE ',I4,I6')                           TAP00090
0005                  READ(15,21) SEG,DATE                                              TAP00100
0006                  WRITE(15,21) SEG,DATE                                             TAP00110
0007             21   FORMAT(I4,I6)                                                     TAP00120
                C                                                                       TAP00130
0008                  DEV=1H                                                            TAP00140
0009                  CALL GETACQ(1,SEG,DATE,ERROR,DEV,'GI')                            TAP00150
0010                  WRITE(16,166) ERROR                                               TAP00160
0011            166   FORMAT('   ERROR',I4)                                           TAP00170
                C                                                                       TAP00180
0012                  CALL WTEERR(ERROR,16)                                             TAP00190
0013                  IF(ERROR .NE. 0 ) STOP                                            TAP00200
0014                  COUNT = 3063                                                      TAP00210
                C                                                                       TAP00220
0015                  CALL TAPEO(1,COUNT,ERROR,BUF)                                     TAP00230
0016                  WRITE(13,11) BUF                                                  TAP00240
0017             11   FORMAT(17(4A4))                                                   TAP00250
0018            226   CONTINUE                                                          TAP00260
0019             55   FORMAT(/,'                          HEADER   BUF')               TAP00270
                C                                                                       TAP00280
0020                  DO 221 I=1,117                                                    TAP00290
0021                  COUNT = 900                                                       TAP00300
0022                  CALL TAPEO(1,COUNT,ERROR,LINE)                                    TAP00310
0023                  WRITE(12,12) LINE                                                 TAP00320
0024             12   FORMAT(15(15A4))                                                  TAP00330
                C     IF(I .LT. 4) WRITE(6,76)                                          TAP00340
0025             76   FORMAT(//,'  LINE')                                              TAP00350
                C     WRITE(6,77) I,IF                                                  TAP00360
0026             77   FORMAT('  ',I0/I0)                                               TAP00370
0027            221   CONTINUE                                                          TAP00380
0028                        STOP                                                        TAP00390
0029                        END                                                         TAP00400
```

FILE TAPCOP PROBLEM / LAWS 3031

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IHCOMH	9C	GFTSCO	A0	RTEERR	A4	TOPRO	A8		

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SEG	FC	DATE	F0	DEV	F4	ERROR	F8	COUNT	FC
I	100								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
BUF	104	LINE	0F4						

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
26	10F2	21	1091	199	10A7	11	10B5	55	10BE
12	10F4	74	10E0	77	10FA				

OPTIONS IN EFFECT [illegible]
OPTIONS IN EFFECT NAME = TAPCOP , LINECNT = 75
STATISTICS SOURCE STATEMENTS = 24,PROGRAM SIZE = 4H22
STATISTICS NO DIAGNOSTICS GENERATED

FILE DSKRED PURDUE / LARS 3031

```
C                                                                       DSK00010
C  DSKRED FFORTRAN          READS DISK FILES AND CONVERTS TO BYTE AND INTDSK00020
C                                                                       DSK00030
0001              INTEGER BUF(765),LINE(225),CH1(196),CH2(196),CH3(196),CH4(196)DSK00040
0002              INTEGER  CPP(196),PURLIN(90),IWO,JOUNG,BIG,BAG        DSK00050
0003              LOGICAL*1 C1(196),C2(196),C3(196),C4(196),PUR(196)    DSK00060
0004              LOGICAL*1  WORD(4),LING(4),BING(4),BANG(4)            DSK00070
0005                 EQUIVALENCE(IWO,WORD(1)),(JOUNG,LING(1))           DSK00080
0006                 EQUIVALENCE (BIG,BING(1)),(BAG,BANG(1))            DSK00090
0007                 EQUIVALENCE(PURLIN(19),PUR(1))                     DSK00100
0008                 EQUIVALENCE(C1(1),LINE(19)) ,(C2(1),LINE(66))      DSK00110
0009                 EQUIVALENCE(C3(1),LINE(113)),(C4(1),LINE(166))     DSK00120
0010                 DATA   30946/517/,BIG/256/,BAG/255/                DSK00130
C                                                                       DSK00140
0011  226         CONTINUE                                              DSK00150
0012              WRITE (16,83)                                         DSK00160
0013  83          FORMAT('  INPUT 1 FOR PURE',/,                       DSK00170
     *                  '       2 FOR MORE PURE',/,                   DSK00180
     *                  '       3 FOR MUST PURE OUTPUT FILE')         DSK00190
0014              READ (15,84) ITEST                                    DSK00200
0015  84          FORMAT(I1)                                            DSK00210
0016              IF(ITEST .LT. 1 .OR. ITEST .GT. 3) GO TO 226          DSK00220
C                                                                       DSK00230
0017              ITEST=ITEST+48                                        DSK00240
0018  77          FORMAT('  ',10I10)                                    DSK00250
0019  999         READ (17,18,END=666) BUF                              DSK00260
0020  18          FORMAT(17I4,A4)                                       DSK00270
0021              WRITE(20,19) BUF                                      DSK00280
C                                                                       DSK00290
C   THIS FINISHES THE HEADER WRITE                                      DSK00300
C                                                                       DSK00310
0022              DO 221 I=1,119                                        DSK00320
0023                 READ(17,12,END=667) LINE                           DSK00330
0024                 READ(17,112,END=668) PURLIN                        DSK00340
0025  112         FORMAT( 90I5A4)                                       DSK00350
0026  12          FORMAT(15(1X,A4))                                     DSK00360
0027              CALL TRNSLT(LINE,PURLIN,CH1,CH2,CH3,CH4,CPP,0)        DSK00370
0028              DO 20 I=1,196                                         DSK00380
C   48 IS MIXED ... 51 IS VERY PURE     HERE WE ZERO 'IMPURE' PIX       DSK00390
0029                 IF(CPP(I) .GE. ITEST) GO TO 20                     DSK00400
0030                    C1(I)=BING(1)                                   DSK00410
0031                    C2(I)=BING(1)                                   DSK00420
0032                    C3(I)=BING(1)                                   DSK00430
0033                    C4(I)=BANG(4)                                   DSK00440
0034  20          CONTINUE                                              DSK00450
C                                                                       DSK00460
C        WRITE(16,69) (CH1(I),CH2(I),CH3(I),CH4(I),CPP(I),I=25,30)      DSK00470
C 69   FORMAT('    ',5I4)                                               DSK00480
C                                                                       DSK00490
0035                 IF (LINE(14) .LT. 4) WRITE(16,62) (LINE(I),I=14,22)DSK00500
0036                 IF (PURLIN(14) .LT. 4) WRITE(16,62) (PURLIN(I),I=14,22)DSK00510
0037                 WRITE(20,12) LINE                                  DSK00520
0038  62          FORMAT('  LINE  ',5Z10)                               DSK00530
0039  221         CONTINUE                                              DSK00540
C                                                                       DSK00550
0040  666         WRITE(16,1166)                                        DSK00560
0041  1166        FORMAT('      END OF DATA ON 17  HEADER')             DSK00570
0042                 STOP                                               DSK00580
0043  667         WRITE(16,1167)                                        DSK00590
0044  1167        FORMAT('  END OF DATA ON 18  SPECTRAL VALUES')        DSK00600
0045                 STOP                                               DSK00610
0046  668         WRITE(16,1168)                                        DSK00620
0047  1168        FORMAT('      END OF DATA ON 19  PURITY')             DSK00630
0048                 STOP                                               DSK00640
0049                 END                                                DSK00650
```

FILE DSKRED PURDUE / LARS 3031

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IHCOM*	F0	IHNSLT	F4						

EQUIVALENCE DATA MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IAM	1 A	AUM3	104	BUM3	10C	BING	10C	BIG	110
BING	110	BIG	114	BUM6	114	PURLIN	11C	PUR	160
LINE	29C	C1	2C8	C2	34C	C3	450	C4	514

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ITEST	6A4	IV	6A8	I	6AC				

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CM1	610	CM1	1204	CM2	1514	CM3	1824	CM4	1B34
CM5	1E44								

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
43	2154	44	2188	77	21AE	14	21B8	112	21C1
12	21C4	62	2193	1169	21E3	1167	2206	1168	222E

OPTIONS IN EFFECT ID,ENCODE,SOURCE,NOLIST,NODECK,NOLOAD,MAP
OPTIONS IN EFFECT NAME = DSKRED , LINECNT = 75
STATISTICS SOURCE STATEMENTS = 43,PROGRAM SIZE = 9614
STATISTICS NO DIAGNOSTICS GENERATED

FILE TRNSLT PIPPINE / LAHS 3031

```
      C TRNSLT FORTRAN
      C  CONVERTS 225 (640) BUFFER TO 4 196 X8=0 CHANNEL LINES                    TRN00010
      C                                                                           TRN00020
      C      SUBROUTINE TRNSLT (DIR,PIN,OH1,OH2,OH3,OH4,PH,OPTION)               TRN00030
      C                                                                           TRN00040
      C                                                                           TRN00050
      C                                                                           TRN00060
      C                                                                           TRN00070
      C                                                                           TRN00080
      C                                                                           TRN00090
      C                                                                           TRN00100
      C                                                                           TRN00110
      C                                                                           TRN00120
      C                                                                           TRN00130
      C                                                                           TRN00140
      C                                                                           TRN00150
      C                                                                           TRN00160
      C                                                                           TRN00170
      C                                                                           TRN00180
      C                                                                           TRN00190
      C                                                                           TRN00200
      C                                                                           TRN00210
      C                                                                           TRN00220
      C                                                                           TRN00230
      C                                                                           TRN00240
      C                                                                           TRN00250
      C                                                                           TRN00260
      C                                                                           TRN00270
      C                                                                           TRN00280
      C                                                                           TRN00290
      C                                                                           TRN00300
      C                                                                           TRN00310
      C                                                                           TRN00320
      C                                                                           TRN00330
      C                                                                           TRN00340
      C                                                                           TRN00350
      C                                                                           TRN00360
      C                                                                           TRN00370
      C                                                                           TRN00380
      C                                                                           TRN00390
      C                                                                           TRN00400
      C                                                                           TRN00410
      C                                                                           TRN00420
      C                                                                           TRN00430
      C                                                                           TRN00440
      C                                                                           TRN00450
      C                                                                           TRN00460
      C                                                                           TRN00470
      C                                                                           TRN00480
      C                                                                           TRN00490
      C                                                                           TRN00500
      C                                                                           TRN00510
      C                                                                           TRN00520
      C                                                                           TRN00530
      C                                                                           TRN00540
      C                                                                           TRN00550
      C                                                                           TRN00560
      C                                                                           TRN00570
      C                                                                           TRN00580
      C                                                                           TRN00590
      C                                                                           TRN00600
      C                                                                           TRN00610
      C                                                                           TRN00620
      C                                                                           TRN00630
      C                                                                           TRN00640
      C                                                                           TRN00650
      C                                                                           TRN00660
      C                                                                           TRN00670
      C                                                                           TRN00680
      C                                                                           TRN00690
      C                                                                           TRN00700
      C                                                                           TRN00710
      C                                                                           TRN00720
      C                                                                           TRN00730
      C                                                                           TRN00740
      C                                                                           TRN00750
      C                                                                           TRN00760
```

LF TRNSLT PURDUE / LARS 3031

```
0064          56      CONTINUE                                                                    TRN00770
                C                                                                                 TRN00780
                C     LANDSAT 3 KAUTH TRANSFORM                                                   TRN00790
0065                  DO 14 K=1,196                                                               TRN00800
0066                  DN1(K)=.5+ .385*CH1(K)+.742*CH2(K)+.842*CH3(K)+.279*CH4(K)                  TRN00810
0067                  DN2(K)=.5 -.324*CH1(K)-.812*CH2(K)+.719*CH3(K)+.412*CH4(K)                  TRN00820
0068                  DN3(K)=.5-1.044*CH1(K)+.527*CH2(K)+.095*CH3(K)-.043*CH4(K)                  TRN00830
0069                  DN4(K)=.5 -.019*CH1(K)+.161*CH2(K)-.563*CH3(K)+.937*CH4(K)                  TRN00840
0070                  PH(K)=PH(K)                                                                 TRN00850
0071          14      CONTINUE                                                                    TRN00860
0072                  RETURN                                                                      TRN00870
0073                  END                                                                         TRN00880
```

FILE TRNSLT PURDUE / LAFS 3031

EQUIVALENCE DATA MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
HHF	C8	IC1	110	C1	110	IC2	104	C2	104
IC3	2C4	C3	2B2	IC4	35C	C4	35C	PUC	44C
ICP	4C4	CP	4A4	CH1	5A4	CC1	5B4	CH2	8C4
CC2	8C4	CH3	8B4	CC3	8A4	CH4	EE4	CC4	EE4
PP	11E4	CCP	11F4	A	1504	IA	1504		

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
J	1508	I	150C	OPTION	1510	K	1514		

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
JHF	1518	JH1	151C	JH2	1520	JH3	1524	JH4	1528
PHF	152C	PH	1530						

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
15	1534								

OPTIONS IN EFFECT ID,EDIT,SOURCE,NOLIST,NODECK,NOLOAD,MAP
OPTIONS IN EFFECT NAME = TRNSLT , LINECNT = 75
STATISTICS SOURCE STATEMENTS = 73,PROGRAM SIZE = 4282
STATISTICS NO DIAGNOSTICS GENERATED

FILE WRTHED PURDUE / LARS 3031

```
0001            SUBROUTINE WRTHED(NCHAN,FEAT,NSAMP,FORMAT,IUNIT)                        WRT00010
0002            IMPLICIT INTEGER (A-Z)                                                  WRT00030
C|*************************************************************************|C|WRT00040
C|                                                                         |C|WRT00050
C|     THE PURPOSE OF TAPWRT IS TO WRITE A DATA TAPE IN EITHER UNIVER-     |C|WRT00060
C|     SAL FORMAT OR LARSYS II FORMAT. THERE ARE TWO ENTRY POINTS TO       |C|WRT00070
C|     THE SUBROUTINE -- WRTHED AND WRTLN .                                |C|WRT00080
C|     WRTHED WRITES THE HEADER RECORD IN 32-BIT BYTES FOR LARSYS II       |C|WRT00090
C|     AND 8-BIT BYTES FOR UNIVERSAL. ONE CALL TO WRTHED MUST BE MADE      |C|WRT00100
C|     FOR EACH REEL OF TAPE. THIS INFORMATION IS PACKED.                  |C|WRT00110
C|                                                                         |C|WRT00120
C|     CALL WRTHED(NC,FEAT,NSAMP,FORMAT,IFORM)                             |  WRT00130
C|          NC     -- NO.OF CHANNELS TO BE WRITTEN FOR EACH DATA SET       |C|WRT00140
C|          FEAT   -- ARRAY CONTAINING CHANNELS TO BE WRITTEN              |C|WRT00150
C|          NSAMP  -- NO. OF SAMPLES PER CHANNEL                           |C|WRT00160
C|          FORMAT -- =1 FOR UNIVERSAL                                     |C|WRT00170
C|          IFORM  -- NO. OF TAPE OUTPUT UNIT                              |  WRT00180
C|                    =2 FOR LARSYS II                                     |C|WRT00190
C|     WRTLN  WRITES THE DATA IN 8-BIT BYTES AND IS ALSO PACKED. A         |C|WRT00200
C|     CALL TO THIS ROUTINE MUST BE MADE FOR EACH DATA SET TO BE WRIT-     |C|WRT00210
C|     TEN                                                                 |C|WRT00220
C|                                                                         |C|WRT00230
C|     CALL WRTLN (IDATA,LSTLIN)                                           |C|WRT00240
C|          IDATA  -- ARRAY CONTAINING DATA TO BE WRITTEN                  |C|WRT00250
C|          LSTLIN -- = 0 FOR N-1 DATA SETS                                |C|WRT00260
C|                    =-1 FOR LAST DATA SET                                |C|WRT00270
C|                                                                         |C|WRT00280
C|          ICHAN  -- ACTIVE CHANNELS HAVE CORRESPONDING BIT POSITION      |C|WRT00290
C|                    TURNED ON                                            |C|WRT00300
C|          PROFLG --                                                      |C|WRT00310
0003            COMMON                                                                  WRT00320
C|               NCS    -- NO. OF CHANNELS                                 |C|WRT00330
C|               NBITS  -- NO. OF BITS PER BYTE                            |C|WRT00340
C|               SVO    -- START OF VIDEO DATA                             |C|WRT00350
C|               DIVF   -- SAME AS NSAMP                                   |C|WRT00360
C|               PRSZ   -- PHYSICAL RECORD SIZE IN BYTES                   |C|WRT00370
C|               NCPR   -- NO. CHANNELS PER RECORD                         |C|WRT00380
C|               NPRC   -- NO. PHYSICAL RECORDS PER CHANNEL                |C|WRT00390
C|               NPRDS  -- NO. OF RECORDS PER DATA SET                     |C|WRT00400
C|               ANCLNG -- LENGTH OF ANCILLARY BLOCK IN BYTES              |C|WRT00410
C|               DUI    -- DATA ORDER INDICATOR                           |C|WRT00420
C|               SAMSTR -- SAMPLE START                                    |C|WRT00430
C|               COMWRD -- SIZE OF COMPUTER WORD IN BITS                   |C|WRT00440
C|               NDSPR  -- NO. OF DATA SETS PER RECORD                     |C|WRT00450
C|               NCAP   -- NO. OF CHANNELS ON ANCILLARY RECORD             |C|WRT00460
C|     PACWAY -- DATA IS PACKED INTO THIS ARRAY AND THEN WRITTEN ON        |C|WRT00470
C|               TAPE BY CALLING WRTLN                                     |C|WRT00480
C|     ICOUNT -- KEEPING TOTAL OF NO. OF DATA SETS WRITTEN                 |C|WRT00490
C|                                                                         |C|WRT00500
C|*************************************************************************|C|WRT00510
0004            LOGICAL*1 PACWAY(3060), ONE, TWO, BIT, IYR                              WRT00520
0005            REAL WAY(700)                                                           WRT00530
0006            DIMENSION VAR(18),PACK(765)                                             WRT00540
0007            DIMENSION IBYTES(18),FEAT(30),IPAY(200)                                 WRT00550
0008             LOGICAL*1 VARIAB(2400)                                                 WRT00560
C                                                                                       WRT00570
C*** COMMON BLOCK CREATED AUG. 3,1979 TO SAVE LARSYS III HEADER                         WRT00580
C                                                                                       WRT00590
0009                COMMON /IOSTOR/ IO0(250)                                            WRT00600
0010            COMMON /WRTAR/ICOUNT,FORMT,IUIT,VARBL(600),IMEM0                        WRT00610
0011            COMMON /TAPEHD/IUNITZ,IFIRST,FSCAN,SAMSTR,SAMDIC,READY,                 WRT00620
               1 NSCAN,LINC,IO(200),OSL,LAJF(30),JPEC(30),IBYTE(30),                    WRT00630
               2 NBUFS,FILENO,LINENO,LININC,NSAMPZ,NCHANZ,FORMTZ                        WRT00640
0012            DATA IBYTES/4,53,90,91,92,96,100,102,103,104,105,107,108,110,           WRT00650
               *      753,1774,1785,1787/                                               WRT00660
0013            DATA ONE/'1'/                                                           WRT00670
0014            DATA NH/4,1,1,1,2,2,2,1,1,1,2,1,2,2,1,1,2,2/                            WRT00680
0015            EQUIVALENCE (VARBL,VARIAB),(PACK,PACWAY),(IWAY,WAY)                     WRT00690
0016            EQUIVALENCE                                    (VARBL(14),NOSAM),       WRT00700
               1            (VARBL(1),ICHAN), (VARBL(7),PRSZ),                       WRT00710
```

FILE WRITER PROBE / (A=5 B=1)

```
                     2          (VARBL(2),VPCFLG),(VARBL(14),NCPW),  (VARBL(15),COMWRD)  WRT00720
                     3,         (VARBL(3),NC),   (VARBL(9),NPRC),                        WRT00730
                     4          (VARBL(4),NBITS),(VARBL(10),NPOS), (VARBL(16),NDSPR),WRT00740
                     5          (VARBL(5),SVD),  (VARBL(11),MCLASS),                     WRT00750
                     6                           (VARBL(12),BUF),  (VARBL(17),NCHR),   WRT00760
                     7          (VARBL(6),NWE),  (VARBL(13),SHIFT),(VARBL(18),NSAM)   WRT00770
0017                 NC = NCHAR                                                         WRT00780
0018                 FORMAT = FMAT                                                      WRT00790
0019                 IUNIT = IUNIT                                                      WRT00800
0020                 ICHAR=1                                                            WRT00810
0021                 SHIFT=1                                                            WRT00820
0022                  ISIGN = 0                                                         WRT00830
0023                 ISTAT =0                                                           WRT00840
0024                 NWE = NSAMP                                                        WRT00850
0025                 NSAM = NSAMP                                                       WRT00860
0026                 NOSAM = NSAMP                                                      WRT00870
0027                 IF ( FORMAT .NE. 1) GO TO 43                                       WRT00880
                C                                                                       WRT00890
                C    ZERO OUT HEADER RECORD STORAGE                                     WRT00900
                C                                                                       WRT00910
0028                 DO 2 I=1,200                                                       WRT00920
0029               2 IARY(I) = IBUF(I)                                                  WRT00930
                C                                                                       WRT00940
                C    PACKING HEADER RECORD IN LAMSYS II FORMAT                          WRT00950
                C                                                                       WRT00960
0030                 IARY(5) =NC                                                        WRT00970
0031                 IARY(6) =NSAMP + 6                                                 WRT00980
                C                                                                       WRT00990
0032                    WRITE = 200                                                     WRT01000
0033                 CALL XWTREC(IUNIT,NOBYTE,IARY)                                     WRT01010
0034                 RETURN                                                             WRT01020
                C                                                                       WRT01030
                C    PACKING HEADER RECORD IN UNIVERSAL FORMAT                          WRT01040
                C                                                                       WRT01050
0035              40 DO 50    I= 1,NC                                                  WRT01060
0036                 II = FFAT(I)                                                       WRT01070
0037              50 ICHAR = ICHAR + 2**(32-II)                                         WRT01080
0038                 NCPW = 1                                                           WRT01090
0039                 PRFLG = 1                                                          WRT01100
0040                 NBITS = 8                                                          WRT01110
0041                 SVD = 1                                                            WRT01120
0042                 PARY = 3060                                                        WRT01130
0043                    PARITY = 3060                                                   WRT01140
0044                 ACLAS = 7                                                          WRT01150
0045                 ANC = ACLAS + 2                                                    WRT01160
0046                 BUF = 0                                                            WRT01170
0047                 CHAR = 32                                                          WRT01180
0048                 NDSPR = 1                                                          WRT01190
0049                 NCPW = 0                                                           WRT01200
0050                 IF (NSAMP .GT. 2999) WRITE(3,510)                                  WRT01210
0051              510 FORMAT(' NO. OF SAMPLES WAS RESET TO 2998')                       WRT01220
0052                 IF (NSAMP .GT. 2998) NSAMP = 2998                                  WRT01230
0053                 ILAST = (NC*NSAMP) + ANC                                           WRT01240
0054                 ILENTH  = (NC*NSAMP + ANC   )/ 3000                                WRT01250
0055                 IJUNK   = MOD((NC*NSAMP + ANC   ),3000)                            WRT01260
0056                 IF (ILENTH .EQ. 0) NCHAR = 3C                                      WRT01270
0057                 IF (ILENTH .EQ. 0) NPOS = 1                                        WRT01280
0058                 IF (ILENTH .EQ. 0) GO TO 80                                        WRT01290
0059                 DO 61    J=1,NC                                                    WRT01300
0060                 IF((NSAMP*J+ANC   ) .GT. 3000)   NCAR = J-1                        WRT01310
0061                 IF((NSAMP*J+ANC   ) .GT. 3000)   GO TO 70                          WRT01320
0062              60 CONTINUE                                                           WRT01330
0063              70 CONTINUE                                                           WRT01340
0064                 NOCHAR = NC - NCAR                                                 WRT01350
0065                 DO 75    J=1,NOCHAR                                                WRT01360
0066                 IF(NSAMP*J .GT. 2998)  NCPW = J-1                                  WRT01370
0067                 IF(NSAMP*J .NE. 2998)  GO TO 76                                    WRT01380
0068              75 CONTINUE                                                           WRT01390
0069                    NCPW = NOCHAR                                                   WRT01400
0070              76 CONTINUE                                                           WRT01410
0071                 NPOPS = NOCHAR / NCPW + 1                                          WRT01420
```

FILE WRTREC PROJECT / LABS 3031

```
0072                IF (MOD(NCHANG,NCPR) .NE. 0)   NRECS = NRECS + 1                    WRT01430
0073              CONTINUE                                                              WRT01440
0074                 IF (ILENTH.NE.0) GO TO 22                                          WRT01450
0075                 PRSZ = (ILEN/1440)*180                                             WRT01460
0076                 IF (PRSZ.NE.ILEN) PRSZ =PRSZ + 180                                 WRT01470
0077           22    CONTINUE                                                          WRT01480
C                                                                                      WRT01490
C       THE  NEXT 3 COMMANDS READ THE HEADER FROM THE HEADER FILE                      WRT01500
C                                                                                      WRT01510
0078                 READ(3) X                                                         WRT01520
0079              READ(35,25)PACK                                                      WRT01530
0080           25 FORMAT(17(4A4))                                                      WRT01540
C                                                                                      WRT01550
C       ZERO OUT PACKAY                                                                WRT01560
C         DO 85  K=1,750                                                               WRT01570
C   85    PACK(K)=0                                                                    WRT01580
C                                                                                      WRT01590
0081              DO 100  K=1,NR                                                       WRT01600
0082              NBYTES = NB(K)                                                       WRT01610
0083              DO 90  L=1,NBYTES                                                    WRT01620
0084              LOC = 4 * L - NBYTES + (K-1)*64                                      WRT01630
0085              BYTE = IBYTES(K)                                                     WRT01640
0086           90 PACKAY(BYTE+L-1) = VARIAB(L,C)                                       WRT01650
0087          100 CONTINUE                                                             WRT01660
C                                                                                      WRT01670
C    THE NEXT 4 COMMANDS PUT THE CURRENT DATE IN THE HEADER                            WRT01680
C    RECORD IN PLACE OF THE ACQUISITION DATE.                                          WRT01690
C                                                                                      WRT01700
0088                CALL IDTE (IMO, IDT, IYR)                                        WRT01710
0089                PACKAY (61) = IMO                                                WRT01720
0090                PACKAY (62) = IDT                                                WRT01730
0091                PACKAY (63) = IYR                                                WRT01740
C                                                                                      WRT01750
C         PACKAY(61) = VARIAB(73)                                                      WRT01760
C         PACKAY(62) = VARIAB(74)                                                      WRT01770
C         PACKAY(63) = VARIAB(75)                                                      WRT01780
C         PACKAY(67) = VARIAB(79)                                                      WRT01790
C         PACKAY(68) = VARIAB(80)                                                      WRT01800
C     THIS ASSIGNS PACKAY(53) TO 0.0                                                   WRT01810
0092              PACKAY(53) = 0.0                                                     WRT01820
C           DO 110 L = 1,512                                                           WRT01830
C  110      PACKAY(111 + L) = VARIAB(111 + L)                                         WRT01840
0093              DO 120 L = 1,15                                                      WRT01850
0094           120   PACKAY(2200 + L) = VARIAB(2200 + L)                             WRT01860
0095                    DO 130 I = 1,4                                                 WRT01870
0096                    IPAT = 2254 + (I - 1)*8                                        WRT01880
0097           130      PACKAY(IPAT) = VARIAB(IPAT)                                   WRT01890
C                                                                                      WRT01900
C***    THE ABOVE THREE LINES IS AN AD HOC ADDITION FOR SOIL LINES                     WRT01910
C***    ADDED OCT. 23,1972                                                             WRT01920
C                                                                                      WRT01930
0098                CALL WRTREC(UNIT,PRSZ(N),PACKAY)                                 WRT01940
0099                RETURN                                                           WRT01950
0100                END                                                              WRT01960
```

FILE WRTHED [illegible] SYMBOLS / LABELS 3031

COMMON BLOCK /[illegible] / MAP SIZE [illegible]

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IDD	0								

COMMON BLOCK /[illegible] / MAP SIZE [illegible]

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ICOUNT	0	FORMT	4	[illegible]LL	8	VARBL	C	VARIAB	C
ICHAN	C	PROFLG	10	IC	14	UNITS	18	SVD	1C
NVE	20	PRSZ	24	[illegible]	28	NARC	2C	NRPDS	30
ARCLNG	34	ODI	38	SAVSTR	3C	NOSAA	40	COMERD	44
NOSPR	44	NCAR	4C	NSAA	50	IREAD	54C		

COMMON BLOCK /TAPERR / MAP SIZE [illegible]

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IUNITZ	0	IFRST	4	FSCNT	8	SAMEND	C	SAMINC	10
READY	14	NSCAN	18	LINE	1C	ID	20	DSL	340
LINF	344	JREC	3PC	IFSTE	434	NBUFS	4AC	FILEND	4B0
LINEND	494	LINING	498	NSAMPZ	49C	NCHANZ	4C0	FORMTZ	4C4

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
#RTREC	11C	FIXPT#	120	IHCDF#	124	IDTE	128		

EQUIVALENCE DATA MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
PACK	168	PACPAY	16A	IPAY	05C	MAY	05C		

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NCHAN	107C	FORMAT	1080	OPBIT	1084	ISTAT	1088	NSAMP	108C
I	1090	NBYTF	1094	II	1098	PRSZRD	109C	ANC	10A0
ILEN	10A4	ILENTH	10A8	J	10AC	NOCHAN	10B0	K	10B4
NBYTES	10B8	L	10BC	LOC	10C0	BYTE	10C4	IPAT	10C8
ONE	10CC	IMO	10D0	IDT	10CE	IYR	10CF		

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NM	1000	INYTES	1114	FFAT	1150				

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
510	1164	25	1168						

OPTIONS IN EFFECT ID,EBCDIC,SOURCE,NOLIST,DECK,NOLOAD,MAP
OPTIONS IN EFFECT NAME = WRTHED , LINECNT = 75
STATISTICS SOURCE STATEMENTS = 108,PROGRAM SIZE = 6200
STATISTICS NO DIAGNOSTICS GENERATED

FILE PSPPAT PURDUE / LARS 3031

```
      SUBROUTINE PSPPAT(MEANS,STDEV,N,CL,IDAT,IPLACE,AVP,AMN,MEN)                    PSP00010
      IMPLICIT INTEGER (A-Z)                                                         PSP00020
        COMMON A(10500)                                                              PSP00030
        DIMENSION L(42,60)                                                           PSP00040
        EQUIVALENCE (LARRAY(1),ARRAY(1))                                             PSP00050
        LOGICAL*1 LPACK(4)                                                           PSP00060
      DIMENSION IPACK(1)                                                             PSP00070
        EQUIVALENCE (IPACK(1),LPACK(1))                                              PSP00080
C                                                                                    PSP00090
C       I CLUSTER CONTROL LIST                                                       PSP00100
C       LCLUS = CLUSTER LIST                                                         PSP00110
      COMMON/PSCNT/ NCL,CNT,MAXPTS,MAXCLS,NLSIN,SEP,                                 PSP00120
     *                              NP,NPTS,   IMAX,  KPTS, NOPTS, PUNCH,          PSP00130
     *  ICMP,CMPIMG,ICMPI(60),MEANS,IMPMIN,                                        PSP00140
     *                  MPCL,MAXCLS,CLSIZE,NOPTS,IPT,ERIMP,TOTPTS,                 PSP00150
     *                   CLASS,ICLS,TOTSUM,TOTSQ,TOTVAR,NOCL,NVAR                  PSP00160
     *  ,KTCLS,NPEAT, ICLS,ECIVEC(30),SYMTX(62)                                    PSP00170
     *,VCMPIZ,STRTEY,ISAVEY,MAPFMT,MAPKEY,OBJOBJ(20),PERLEN,SIMERR                  PSP00180
     *,IMAGE,IMAGIT,IMFILE,INIT,MAPIN,SIMVEC(62),NOSUMS,CHNVC(30)                  PSP00190
     *  ,CMAP,ESP(MAP,,,KEAN,MEANS),MEANS),                                       PSP00200
     *  SP,SYMBL,ITRIGR,ITRIGU,COFLAG,                                             PSP00210
     *  ,IRL36,OOB,SLOPTS(60),NSLOTS,SUNCOM(30),LENCAT,                            PSP00220
     *  TYPE(650,2),,,SCI(10),C),IVPAT(11,2),IUCAT(2),NOUT(2)                      PSP00230
     *  ,,TEST),MAXPTS                                                             PSP00240
      REAL MEANS                                                                     PSP00250
      COMMON/ISCLAS/SCLAS(65),ISCAT,ISUMC,SMSTR,SMSTP,SMINC,LINSKP                  PSP00260
C$ENT                                                                              PSP00270
      DIMENSION IPLACE(NPTS),AVP(NOFEAT,MAXCLS)                                      PSP00280
      DIMENSION STDEV(NOFEAT,MAXCLS),CL(MAXCLS,MAXCLS),N(MAXCLS)                     PSP00290
      DIMENSION MEANS(NOFEAT,MAXCLS)                                                 PSP00300
C   THE NEAT LIBRARY WAS CHANGED TO AVOID PROPOGATE PROBLEMS                       PSP00310
      REAL*8 ARRAY(15,60),VAR                                                        PSP00320
      DIM  MEAN(NOFEAT,MAXCLS)                                                       PSP00330
      REAL MEAN,STDEV,SUM,SQUST,DIST,RND,MEANS                                       PSP00340
      DIMENSION CSUM(11)                                                             PSP00350
      REAL CSUM                                                                      PSP00360
      REAL AMN,AMN                                                                   PSP00370
        IM = (IDAT) - 11**                                                           PSP00380
        IPACK(1) = 0                                                                 PSP00390
        ,,, = ,00000                                                                 PSP00400
C                                                                                    PSP00410
      IF(NOFLAG.NE.0) N(ICAT+1)=NOU(0)OFLAG                                          PSP00420
      IF(NOFLAG.NE.0) N(ICAT+OUNO)=NOU(OOUO)                                         PSP00430
      DO 5 I=1,L (2)                                                                 PSP00440
        ,(I)=0                                                                       PSP00450
      DO J=1,NOFEAT                                                                  PSP00460
      AVP(J,I)=0.0                                                                   PSP00470
      STDEV(J,I)=0.0                                                                 PSP00480
      CL(J,I) = 0.0                                                                  PSP00490
      MEAN(J,I)=0.0                                                                  PSP00500
    5 ARRAY(J,I)=0.0                                                                 PSP00510
C                                                                                    PSP00520
C       ASSIGN DATA TO CLUSTERS                                                      PSP00530
C                                                                                    PSP00540
      ANCLS=NOFEAT-1                                                                 PSP00550
      ANPTS=NPTS                                                                     PSP00560
      ICCT=NPTS                                                                      PSP00570
      I=NPTS                                                                         PSP00580
   20 IF(I.LE.1)ICCI=NPTS                                                            PSP00590
      IF(I.GE.0.1)GO TO 1                                                            PSP00600
      I=RSET(ICCT                                                                    PSP00610
        I=I+1                                                                        PSP00620
        IF (4*ISAVE.GE.IMAGE) I=I+1 = I+1                                            PSP00630
        CALL GETAR(10,ESI,ARRAY(IDAT)),I=I,ISTAT)                                   PSP00640
      ESI=ESI-0.5                                                                    PSP00650
   25 IF(ESI.LE.0.1)GO TO 25                                                         PSP00660
      IF(ISTAT.EQ.0)GO TO 30                                                         PSP00670
      IF(ISTAT.NE.10)GO TO 40                                                        PSP00680
      WRITE(6,30)ISTAT                                                               PSP00690
   30 FORMAT(' ERROR READING DRUM---ISTAT=',I4)                                      PSP00700
   40 CONTINUE                                                                       PSP00710
```

FILE PSPPAT POPCODE / LAPS 30.81

```
0051                IF (ISORT.EQ.0.AND.ISUNC.EQ.0) GO TO 50                                 PSP00720
0052                DO 49 I=1,ICCT                                                          PSP00730
0053                  IBASE = (I - 1)*NOFEAT + 1                                            PSP00740
0054                IF(NOFEAT.LT.2) GO TO 42                                                PSP00750
0055                DO 41 K=2,NOFEAT                                                        PSP00760
0056                  LOCAT(4) = LOCARAY(IBASE + K)                                         PSP00770
0057                  CSUN(K) = LOCAT(1)                                                    PSP00780
0058                IF (CSUN(K).EQ.0.AND.CSUN(K).NE.9999) GO TO 42                          PSP00790
0059          41    CONTINUE                                                                PSP00800
0060                IF (CSUN(1).EQ.9999) IPLACE(I) = LNCAT + 1                              PSP00810
0061                IF (CSUN(1).EQ.0000) IPLACE(I) = LNCAT + 0000                           PSP00820
0062                GO TO 49                                                                PSP00830
0063          42    KK=1                                                                    PSP00840
0064                SQIST=1.0E+20                                                           PSP00850
0065                DO 45 J=1,LNCAT                                                         PSP00860
0066                DIST=0.                                                                 PSP00870
0067                DO 44 K=2,NOFEAT                                                        PSP00880
0068                  LOCAT(4) = LOCARAY(IBASE + K)                                         PSP00890
0069                  CSUN(K) = LOCAT(1)                                                    PSP00900
0070          44    DIST=DIST+SIGN(1.0,MEANS(K,J)-CSUN(K))*SUNCOR(K)                        PSP00910
0071                IF (DIST - SQIST) 45,45,45                                              PSP00920
0072          45    KK=J                                                                    PSP00930
0073                SQIST=DIST                                                              PSP00940
0074          46    CONTINUE                                                                PSP00950
0075          47    CONTINUE                                                                PSP00960
0076                NUM(KK)= NUM(KK)+1                                                      PSP00970
0077                IPLACE(I)=KK                                                            PSP00980
0078                DO 48 K=2,NOFEAT                                                        PSP00990
0079                SUM(KK,K)=SUM(KK,K)+CSUN(K)                                             PSP01000
0080                  SUMSQ(KK,K)=SUMSQ(KK,K)+CSUN(K)                                       PSP01010
0081                AVG(KK,K)=AVG(KK,K)+CSUN(K)**2                                          PSP01020
0082                  DAVG(KK,K)=DAVG(KK,K)+CSUN(K)**2                                      PSP01030
0083          48    CONTINUE                                                                PSP01040
0084          49    CONTINUE                                                                PSP01050
0085                GO TO 101                                                               PSP01060
0086          50    DO 100 I = 1,ICCT                                                       PSP01070
0087                  IBASE = (I - 1)*NOFEAT + 1                                            PSP01080
0088                  K=1                                                                   PSP01090
0089                IF(NOFEAT.EQ.1) GO TO 52                                                PSP01100
0090                DO 51 K=2,NOFEAT                                                        PSP01110
0091                  LOCAT(4) = LOCARAY(IBASE + K)                                         PSP01120
0092                  CSUN = LOCAT(1)                                                       PSP01130
0093                IF (CSUN.EQ.9999.AND.CSUN.NE.0000) GO TO 52                             PSP01140
0094          51    CONTINUE                                                                PSP01150
0095                IF (CSUN.EQ.9999) IPLACE(I) = LNCAT + 1                                 PSP01160
0096                IF (CSUN.EQ.0000) IPLACE(I) = LNCAT + 0000                              PSP01170
0097                GO TO 100                                                               PSP01180
0098          52    CONTINUE                                                                PSP01190
0099                KK = 1                                                                  PSP01200
0100                SQIST=1.0E+20                                                           PSP01210
0101                DO 70 J=1,LNCAT                                                         PSP01220
0102                DIST=0.0                                                                PSP01230
0103                DO 55 K = 1,NOFEAT                                                      PSP01240
0104                  LOCAT(4) = LOCARAY(IBASE + K)                                         PSP01250
0105                  CSUN(K) = LOCAT(1)                                                    PSP01260
0106          53    DIST = DIST + ABS(MEANS(K,J) - CSUN(K))                                 PSP01270
0107                IF(DIST-SQIST)60,70,70                                                  PSP01280
0108          60    KK=J                                                                    PSP01290
0109                SQIST=DIST                                                              PSP01300
0110          70    CONTINUE                                                                PSP01310
0111          80    CONTINUE                                                                PSP01320
0112                NUM(KK)=NUM(KK)+1                                                       PSP01330
0113                IPLACE(I)=KK                                                            PSP01340
0114                DO 90 K=1,NOFEAT                                                        PSP01350
0115                SUM(KK,K) = SUM(KK,K) + CSUN(K)                                         PSP01360
0116                SUMSQ(KK,K) = SUMSQ(KK,K) + CSUN(K)                                     PSP01370
0117                AVG(KK,K) = AVG(KK,K) + CSUN(K)**2                                      PSP01380
0118                DAVG(KK,K) = DAVG(KK,K) + CSUN(K)**2                                    PSP01390
0119          90    CONTINUE                                                                PSP01400
0120         100    CONTINUE                                                                PSP01410
0121         101    CONTINUE                                                                PSP01420
```

FILE PSPRAT PURDUE / LARS 3031

```
0122                  IF(IRD.EQ.0)GO TO 110                                    PSP01430
0123                  CALL RANWIT(ASW=S2,IPLACE,ICCT,ISTAT)                    PSP01440
0124                  AQSFS2=AQSFS2+ICCT                                       PSP01450
0125           105    IF(ISTAT.EQ.1)GO TO 105                                  PSP01460
0126           110    IRC=IRC-1                                                PSP01470
0127                  IF(IRC.ST.0)GO TO 20                                     PSP01480
0128                    KA = 1                                                 PSP01490
0129            115     CONTINUE                                               PSP01500
0130                  DO 130 K=1,LAST                                          PSP01510
0131                 IF(N(K).EQ.0.)GO TO 130                                   PSP01520
0132                  RNO=FLOAT(N(K))                                          PSP01530
0133                  DO 130 J=1,JFEAT                                         PSP01540
0134                  AMN(J,K)=AMN(J,K)/RNO                                    PSP01550
0135                            DAMN(J,K)=DAMN(J,K)/RNO                        PSP01560
0136                  MEANS(J,K)=AMN(J,K)                                      PSP01570
0137                            VAR = (SSAVP(J,K)/RNO-DAMN(J,K)*DAMN(J,K))     PSP01580
0138                   IF(VAR .LT. .0001) VAR=.0001                            PSP01590
0139                  STDEV(J,K)=SQRT(VAR)                                     PSP01600
0140                   DUMA = STDEV(J,K)                                       PSP01610
0141                   IF (DUMA.LT..001)    STDEV(J,K) = 0.0                  PSP01620
0142           130     CONTINUE                                               PSP01630
0143                   RETURN                                                 PSP01640
0144                   END                                                    PSP01650
```

FILE PGMNAM PGMNAM / IBCOM 0031

COMMON BLOCK / / NAME SIZE 0040

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ARRAY	0								

COMMON BLOCK /PGMCM/ / MAP SIZE 0040

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NTOP	0						C	STOMAX	10
NLMIN	4				8C		20		
NSTK	8				90				
IOMSG	0C				94				
LPAGE	144				98				
LSTOTS	18				9C				
TOTAL	14C				A0				
MAXLLS	1C				A4				
IOMSY	2C				A8				
SIZE	34				AC				
DATE	38				44C				
PROGID	4C				4C4				
SYMID	5C				5C8				
NPGM	58				5C4				
NVOT	4C				1C38		1C		1C38
OFFSET	1040	1044							

COMMON BLOCK /IS [illegible] / MAP SIZE 38

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CURSOR	0				24		2C		2C
LPTNC	28								

SYMBOLIC LABEL

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
DREAD	1FC								

EXTERNAL DATA REF

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IPACK	2C								

SCALAR

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
V	2C	I	21C	[illegible]	224	DJ	21C	I	21C
J	22C	[illegible]	224	[illegible]	228	ICCR	22C	IHC	230
IRECS	234	I	234	INTAT	238	IHAXE	240	F	244
CD10	24C	[illegible]	24C	SWIDT	250	WIDT	254	KA	258
MAT	25C	DMAE	264						

ARRAY REF

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IPLACE	26C	AM	264	SIMEO	[illegible]	CLJ	270	N	274
ANC	274	[illegible]	278	HAVE	[illegible]	SIMM	2000	MEM	3E80
CSIM	3E84								

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
30	3EFC								

*OPTIONS IN *EFFECT* [illegible]
OPTIONS IN EFFECT NAME = PGMNAM , LINECNT = 75
STATISTICS SOURCE STATEMENTS = 144,PROGRAM SIZE = 19996
STATISTICS NO DIAGNOSTICS GENERATED

FILE COVPAT PROBE / CAPS 3031

```
0001          SUBROUTINE COVPAT(COVAR,IDAT1,IPLACE,MEANS,N,IHAD)                    COV00010
C               OSCAR ANDERSEN MAY DAY EIGHTY                                       COV00020
C               CM 4 > 250 ... COV = SQI                                            COV00030
0002          IMPLICIT INTEGER (A-Z)                                                COV00040
0003                COMMON SOMAT(10000)                                             COV00050
0004                LOGICAL*1 LARRAY(4200)                                          COV00060
0005                EQUIVALENCE (SMMAT,LARRAY)                                      COV00070
0006                LOGICAL*1 LPACK(4)                                              COV00080
0007                EQUIVALENCE (LPACK,IPACK)                                       COV00090
C*                                                                                  COV00100
C*    SUBROUTINE COVAR CALCULATES AND PRINTS THE COVARIANCE MATRIX FOR              COV00110
C*    EACH CLUSTER                                                                  COV00120
C*                                                                                  COV00130
C     INCLUDE (COMMON.LIST                                                          COV00140
C     INCLUDE (DIMENSION.LIST                                                       COV00150
0008          COMMON/PASS/NUMVAR,IDAT,NVI(4),KK,SIGMA,BLANK,SEP,                    COV00160
     *                    MEP,NPTS(10),   LAB,  EPIS, NUMPTS, PUNCH,              COV00170
     *  IC4,CHARS,IC(4),(2),MASK,IBEGI,IEGI(4),                                     COV00180
     *                 PROMPT,RESP,LIS,CARD,DELM,IPI,TOTAND,TOTPTS,                 COV00190
     *                 VOLASS,VCLS,IFSUM,IPTFLG,TITVAT, IDLE, SMT                   COV00200
     * ,VARTOL,STATV,IDENT,VMEAN,SMEAN,NCLUS(30),SCATA(64)                          COV00210
     *,VARSIZ,STATV,IDENT,VMEAN,SMEAN,NCLUS(20),MEANCH,SIGMSQ                      COV00220
     *,IDENT,IMAGE,IMAGE,IMAGE,IMAGE,SUMVEC(100),MCSUMS,CHIVC(30)                  COV00230
     * ,IDCHAR,FREQ(4),MEAN,MEANSQ,                                                COV00240
     * SC(30),SC(30),ITYPE,ITRIG,DOFLAG,                                           COV00250
     * DOFLAG,NUMCLS,IPTS(50),ASUMS,SUMSQ(30),LLNCAT,                              COV00260
     * NVECTP(20),NPCT(20,2),NVAT(11,2),IDCAT(2),MOD(2)                            COV00270
     * ,XEPT),MEANSQ                                                               COV00280
0009          REAL SUMSQ                                                          COV00290
0010          COMMON/BLANK/MEAN(64),,APTAB,DATABE,SAVTAB,DAFILE,MMKEY,            COV00300
     *                 MISFIL,MISKEY,THROPM,FRIPTP,EPPKEY,MAPUNT,MUFILE,          COV00310
     *    MPUNCH,MPRINT,PASSIZ,MAFIL,SIGFIL,ASAV,ASAVFL                            COV00320
     *  ,MMSTAT,MMSTFI,SCRTCH,,MAPFIL                                              COV00330
     *  ,STATNT,STATFL,,CARDS,IMAGFL,SMTAFL,MISTFL,MCHUNT,                         COV00340
     *  CHPUNT,PRINT,MAXDIM                                                       COV00350
      $SENT                                                                       COV00360
0011          REAL MEANS,COVAR,TOL,SUM(50),LET                                    COV00370
0012          DIMENSION COVAR(MAPSIZ,1,CAT),IPLACE(NOPTS)                         COV00380
0013          DIMENSION MEANS(NOPEAT,MAXCLS),N(MAXCLS)                            COV00390
0014          DATA CH/'CM'/                                                       COV00400
0015          DO 10 I=1,NCAT                                                      COV00410
0016          DO 10 J=1,MAPSIZ                                                    COV00420
0017       10 COVAR(J,I)=0.0                                                      COV00430
C*                                                                                COV00440
0018               READ 900                                                       COV00450
0019               IH = (IDAT1 - 1)*4                                             COV00460
0020               IPACK = IH                                                     COV00470
0021            IMAG=1                                                            COV00480
0022            IDLE=.NOT.MAPPI                                                   COV00490
0023            AD=ESI=MSIAT                                                      COV00500
0024            AD=ESQ= MFMIAT                                                    COV00510
0025            ICCT=NPTS                                                         COV00520
0026            IHC=IH                                                            COV00530
0027       20 IF(IPC.LE.1)ICCT=KPTS                                               COV00540
0028          IF (IHC.EQ.0) GO TO 30                                              COV00550
0029          IWD=ICCT*NOFEAT                                                     COV00560
0030               IWD4 = I*BYS/4                                                 COV00570
0031               IF (4*IWD4.NE.IWBYS) IWD4 = IWD4 + 1                           COV00580
0032            CALL SREAD(A(0)=ESI,AMEAN(IDAT1),IWD4,ISTAT)                      COV00590
0033          AD=ESI=AD=ESI+IWD4                                                  COV00600
0034       22 IF(ISTAT.EQ.1)GO TO 22                                              COV00610
0035          CALL MREAD(AMESQ,IPLACE,ICCT,ISTAT)                                 COV00620
0036          AD=ESQ=AD=ESQ+ICCT                                                  COV00630
0037       25 IF(ISTAT.EQ.1)GO TO 25                                              COV00640
C*                                                                                COV00650
C*    SINCE THE COVARIANCE MATRIX IS SYMMETRICAL ONLY THE LOWER                   COV00660
C*    TRIANGULAR PORTION OF THE MATRIX IS CALCULATED.                             COV00670
C*                                                                                COV00680
0038        30    DO 45 I = 1,ICCT                                                COV00690
0039                  IMASK = IH + (I - 1)*NOFEAT                                 COV00700
0040               KK=0                                                           COV00710
```

FILE COVPAT MPMODE / LANS 3031

```
0041                ICLS=IPLACE(I)                                                  COV00720
0042                IF(ICLS.GT.LNCAT) GO TO 45                                      COV00730
0043                DO 40 J=1,NOFEAT                                                COV00740
0044                DO 40 K=1,J                                                     COV00750
0045                   LPACK(4) = LARRAY(IBASE + J)                                 COV00760
0046                   IPACK1 = IPACK                                               COV00770
0047                   LPACK(4) = LARRAY(IBASE + K)                                 COV00780
0048                   IPACK1 = IPACK1*IPACK                                        COV00790
0049                   IPACK = IPACK1                                               COV00800
0050                KK=KK+1                                                         COV00810
0051                COVAR(KK,ICLS)=COVAR(KK,ICLS)+DUM                               COV00820
0052             40 CONTINUE                                                        COV00830
0053             45 CONTINUE                                                        COV00840
0054                I=C=I-1                                                         COV00850
0055                IF (I.GT.1) GO TO 20                                            COV00860
0056                DO 50 I=1,LNCAT                                                 COV00870
0057                IF(N(I).EQ.1)GO TO 50                                           COV00880
0058                KK=1                                                            COV00890
0059                DO 51 J=1,NOFEAT                                                COV00900
0060                DO 51 K=1,J                                                     COV00910
0061                KK=KK+1                                                         COV00920
0062                COVAR(KK,I)=COVAR(KK,I)/N(I) - MEANS(K,I)*MEANS(J,I)            COV00930
                 C                                                                  COV00940
                 C   THE NEXT 2 STATEMENTS WERE INSERTED FOR IMPURE PIXEL COVARIANCE COV00950
                 C                                                                  COV00960
0063                IF(VARS(4,I) .GT. .25) COVAR(KK,I)=0.01                         COV00970
0064                IF(K .EQ. J.AND. MEANS(4,I).GT.2.5)) COVAR(KK,I)=5.0            COV00980
0065             50 CONTINUE                                                        COV00990
0066                IACEPT=PMIN*NOFEAT                                              COV01000
0067                IF(IACEPT.LT.NOFEAT)GO TO 58                                    COV01010
                 C                                                                  COV01020
                 C     CHECK FOR SINGULAR COVARIANCE MATRIX                         COV01030
                 C                                                                  COV01040
0068                DO 51 I=1,LNCAT                                                 COV01050
0069                CALL CHOLET(COVAR(1,I),NOFEAT,DUMP,DET)                         COV01060
0070                IF(DET.LT.TRL)GO TO 52                                          COV01070
0071             51 CONTINUE                                                        COV01080
0072                GO TO 58                                                        COV01090
                 C                                                                  COV01100
                 C      DELETE SINGULAR COVARIANCE CLUSTER                          COV01110
                 C                                                                  COV01120
0073             52 WRITE(6,150)I                                                   COV01130
0074                IF(LNCAT.EQ.1)CALL CHERR                                        COV01140
0075                ISAD=1                                                          COV01150
0076                LNCAT=LNCAT-1                                                   COV01160
0077                LLNCAT=LLNCAT-1                                                 COV01170
0078                DO 53 II=I,LNCAT                                                COV01180
0079                DO 53 III=1,NOFEAT                                              COV01190
0080                MEANS(III,II)=MEANS(III,II+1)                                   COV01200
0081             53 CONTINUE                                                        COV01210
0082             54 RETURN                                                          COV01220
0083            150 FORMAT(2X,'CLUSTER',I5,' DELETED FOR SINGULARITY')              COV01230
                 C  IF(STATKY.EQ.1)RETURN                                           COV01240
0084             55 WRITE (6,HEAD)                                                  COV01250
0085                WRITE(6,150)CLS,AM                                              COV01260
0086                DO 70 I=1,LNCAT                                                 COV01270
0087                WRITE (6,90)I                                                   COV01280
0088                DO 70 LOC=1,NOFEAT,12                                           COV01290
0089                ISTOP=LOC+11                                                    COV01300
0090                IF(ISTOP.GT.NOFEAT)ISTOP=NOFEAT                                 COV01310
0091                WRITE(6,140)(CHEFVEC(J),J=LOC,ISTOP)                            COV01320
0092                II=1                                                            COV01330
0093                KINC=1                                                          COV01340
0094                DO 60 J=LOC,NOFEAT                                              COV01350
0095                K=J*(J+1)/2-II+1                                                COV01360
0096                JK=K+KINC-1                                                     COV01370
0097                WRITE(6,100)(COVAR(M,I),M=K,JK)                                 COV01380
0098                II=II+1                                                         COV01390
0099             60 IF(KINC.LT.ISTOP.AND.KINC.LT.12)KINC=KINC+1                     COV01400
0100                WRITE (6,110)                                                   COV01410
0101             70 CONTINUE                                                        COV01420
```

FILE COVPAT

```
0102          80 CONTINUE                                                          COV01430
0103             RETURN                                                            COV01440
0104          90 FORMAT(///' COVARIANCE MATRIX FOR CLUSTER',I4/)                   COV01450
0105         100 FORMAT(/6X,12F9.2)                                                COV01460
0106         110 FORMAT(///)                                                       COV01470
0107         120 FORMAT(1H1)                                                       COV01480
0108         140 FORMAT(9X,12(A3,I2,')',3X))                                       COV01490
0109         150 FORMAT(/' COVARIANCES FOR CLASS',2X,A4//)                         COV01500
0110             END                                                               COV01510
```

FILE COVPAT PURDUE / CAMS 3031

COMMON BLOCK / / MAP SIZE 45A8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ARRAY	0	LARRAY	8						

COMMON BLOCK /PASS / MAP SIZE 10A8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
STOP	0	LOCAT	4	NPTS	8	KRR	C	STOMAX	10
DLMIN	14	SEP	18	MAP	1C	SPTHIS	20	IMD	24
KPTS	28	NPTS	2C	PTCH	30	ICMX	34	CMSIMS	38
ICHASH	3C	[illegible]	134	[illegible]	138	MEGIN1	13C	MEGIN2	140
BEGIN3	144	CLSMEM	148	NOFLD	14C	IPI	150	TOTMRD	154
TOTMIS	158	NCLASS	15C	NOCLS	160	TOTSUM	164	TOTFLD	168
IDENPT	16C	NOFL	170	NPRT	174	MATC_S	178	NOFEAT	17C
MAXCLS	180	FFTYPE	184	SYMMTX	1FC	VARSIZ	2F4	STATRY	2FB
ISCKEY	2FC	MSPEAT	300	MSPKEY	304	SEMMEM	308	PERCEN	358
SIMESP	35C	IDMSP	360	IDSMII	364	IMFILE	368	INITM	36C
PMIN	370	SIMMRC	374	NJSQRT	48C	CMAVC	470	NOCMAN	4E8
EVCOMP	4EC	NOSEP	4F0	MEANSQ	4F4	MEANSQ	4F8	SYMSQ	4FC
SYMSQ	500	ITHIRD	504	ITHIRD	508	DOFLAG	50C	DOFLAG	510
DOSQ	514	STBOTS	518	NSQRTS	804	SUMCOM	80C	LL_CAT	884
COVMAT	888	DIFCT	E58	NVPRT	1038	IOCNT	1090	NDSQ	1098
MXFET1	10A8	MXFET2	10A4						

COMMON BLOCK /GLOBAL / MAP SIZE 1B0

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
HEAD	0	MAPTAP	FC	MATAPE	100	SAVTAP	104	BMFILE	108
MMKEY	10C	MISFIL	110	MISKEY	114	IMFORM	118	EMIMTP	11C
FSSKEY	120	MIMPUT	124	MUFILE	128	CMMAIN	12C	NMMOUS	130
PAGSIZ	134	DBFILE	138	STAFIL	13C	ASAV	140	ASAVFL	144
NMSTOR	148	MMSTFI	14C	SCTMOR	150	TMPSFL	154	MAPFIL	158
BUTFIL	15C	MCMPAS	160	TMPSFL	164	MAPFIL	168	DJTUNT	16B
PCMOUT	170	CMBUNT	174	PMTUNT	178	PAMOTO	17C	HISTFL	18C

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
RREAD	1EC	CHLORT	150	IBCOM#	164	CMERR	168		

EQUIVALENCE DATA MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
LPACK	1A0	IPACK	1A8						

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CM	1A4	I	1A8	J	1AC	IM	1B0	IUNIT	1B4
IMAD	1B5	TJL	1BC	ADDRES1	1C0	ADDRES2	1C4	ICCT	1C8
IPC	1CC	IWORDS	1D0	I1M4	1D4	ISTAT	1D8	IMASE	1DC
KK	1E0	IGLS	1E4	*	1E8	IPASS1	1EC	OUM	1F0
IACEPT	1F4	OPT	1F8	II	1FC	III	200	LOC	204
ISTOP	208	KIND	20C	JK	210	*	214		

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
DIAM	218	COVAR	308	IDPACE	30C	MEANS	310	M	314

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
160	314	90	342	100	364	110	373	120	379
140	370	150	380						

OPTIONS IN EFFECT IOPT=EBCDIC,SOURCE,NOLIST,NODECK,NOLOAD,MAP
OPTIONS IN EFFECT NAME = COVPAT , LINECNT = 75
STATISTICS SOURCE STATEMENTS = 111,PROGRAM SIZE = 3484

FILE COVPAT

STATISTICS NO DIAGNOSTICS GENERATED

FILE IDTE

```
                      SUBROUTINE IDTE (IDA, IMO, IYC)                                   IDT00010
C                                                                                       IDT00020
C     THIS SUBROUTINE OBTAINS THE CURRENT DATE ( FROM FX  JUNE 17,1980 )                IDT00030
C     AND CONVERTS IT TO A 3 BYTE REPRESENTATION ( FROM EX 6 17 80 ).                   IDT00040
C                                                                                       IDT00050
C                                                                                       IDT00060
      LOGICAL*1 IDATE(4), NDATE(4), IDA, IMO, IYC                                       IDT00070
      EQUIVALENCE (IDATE,NDATE), (NDATE(1),IDA), (NDATE(2),IMO), (NDATE(3),IYC)         IDT00080
      DIMENSION IDATE(3), IBUF(2)                                                       IDT00090
C                                                                                       IDT00100
      CALL GDATE(IDATE)                                                                 IDT00110
      ENCODE (12,32,IBUF) IDATE                                                         IDT00120
   32 FORMAT (3A4)                                                                      IDT00130
      DECODE (32,33,IBUF)                                                               IDT00140
      ENCODE (32,33,IBUF)                                                               IDT00150
   33 FORMAT (5X,I2,3X,I2)                                                              IDT00160
C                                                                                       IDT00170
      CALL IBYTE (IDATE(1),J)                                                           IDT00180
      IDA = IDATE(2)                                                                    IDT00190
      IMO = NDATE(4)                                                                    IDT00200
      IYC = NDATE(3)                                                                    IDT00210
      RETURN                                                                            IDT00220
      END                                                                               IDT00230
                                                                                        IDT00240
```

FILE IDTE [illegible] BUBBLE / LABS 3631

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
GTDATE	90	IBCOM#	94	IMONTH	98				

EQUIVALENCE DATA MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IGOOD	A8	NUDATE	A8	NUDTE	B0	J	B0		

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NA	B4	NB	B5	NC	B6				

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
DATE	B8								

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
32	C4	33	C4						

OPTIONS IN EFFECT ID,EBCDIC,SOURCE,NOLIST,DECK,NOLOAD,MAP
OPTIONS IN EFFECT NAME = IDTE , LINECNT = 75
STATISTICS SOURCE STATEMENTS = 15,PROGRAM SIZE = 524
STATISTICS NO DIAGNOSTICS GENERATED

FILE IMONTH PURDUE / LARS 3031

```
                C                                                                IM000010
0001                SUBROUTINE IMONTH (DATE, I)                                  IM000020
                C                                                                IM000030
                C   THIS SUBROUTINE CHANGES THE MONTH FROM ALPHA CHARACTERS TO    IM000040
                C   NUMERIC CHARACTERS.                                          IM000050
                C                                                                IM000060
0002                DIMENSION TEST(12)                                           IM000070
0003                DATA TEST/'JAN','FEB','MAR','APR','MAY','JUNE','JULY',       IM000080
                 1 'AUG','SEPT','OCT','NOV','DEC'/                               IM000090
                C                                                                IM000100
0004                DO 20  I = 1,12                                              IM000110
0005                IF (DATE .EQ. TEST(I))  RETURN                               IM000120
0006          20    CONTINUE                                                     IM000130
0007                WRITE(6,45)                                                  IM000140
0008          45    FORMAT (43H PROGRAM MONTH HAS NOT MATCHED A MONTH)           IM000150
0009                RETURN                                                       IM000160
0010                END                                                          IM000170
```

FORTRAN IV G LEVEL 21 [illegible] 11/31/12 PAGE 0002

FILE IMONTH

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IPCOM#	83						
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
I	CC						
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TEST	A4						
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
45	64						

OPTIONS IN EFFECT [illegible]
OPTIONS IN EFFECT [illegible]
STATISTICS SOURCE STATEMENTS = [illegible] PROGRAM SIZE =
STATISTICS NO DIAGNOSTICS GENERATED